Lista 1.2, aula 3. Tópico: Verossimilhança

Henrique Ap. Laureano
laureano@ufpr.br $\wedge$ www.leg.ufpr.br/~henrique/

Contents

<r code>

```r
# packages =============================================================
library(numDeriv)
library(ggplot2)
library(wesanderson)
library(fishualize)
library(latex2exp)
library(gridExtra)
```

Exercise 1: dist. Poisson

Seja X uma v.a. de uma distribuição de Poisson ($X \sim P(\lambda)$) para a qual foi obtida a seguinte amostra aleatória: (3, 1, 0, 2, 1, 1, 0, 0).

<r code>

```r
data_1 <- c(3, 1, 0, 2, 1, 1, 0, 0)
```

(a) Obtenha a função de verossimilhança, sua aproximação quadrática e intervalos de confiança (pelo menos duas formas) para λ.

The likelihood and the log-likelihood of a Poisson(λ) is given by

$$L(\lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \quad \text{and} \quad \log L(\lambda) = l(\lambda) = -n\lambda + \log\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \log x_i!.$$

We work here with the log-likelihood to make our life easier, since the computations are much simpler in the log scale.

By computing the Score function (derivative of $l(\lambda)$ wrt to λ) and making it equal to zero, we find the MLE $\hat{\lambda} = \bar{x}$. Computing the second derivative we find the observed information, $I_O(\lambda)$

$$I_O(\lambda) = n\bar{x}/\lambda^2 \quad \rightarrow \quad I_O(\hat{\lambda}) = n/\bar{x}.$$

A graph of the Poisson log-likelihood is provided in Figure 1.

Doing a quadratic approximation (a Taylor expansion of second order) in $l(\lambda)$ around $\hat{\lambda}$, we have

$$\begin{aligned} l(\lambda) &\approx l(\hat{\lambda}) + (\lambda - \hat{\lambda})l'(\hat{\lambda}) + \frac{1}{2}(\lambda - \hat{\lambda})^2 l''(\hat{\lambda}) \\ &= l(\hat{\lambda}) + \frac{1}{2}(\lambda - \hat{\lambda})^2 l''(\hat{\lambda}) \\ &= l(\hat{\lambda}) + \frac{1}{2}(\lambda - \hat{\lambda})^2 I_O(\hat{\lambda}), \quad \text{since the Score is zero at the MLE.} \end{aligned}$$

A graph of the quadratic approximation of the Poisson log-likelihood is also provided in Figure 1. There we can see how good is the approximation around the maximum likelihood estimator (MLE). Here the sample size is very small, the idea is that as the sample size increase, the quality, in this case the range, of the approximation also increase. This should happen because as the sample size increase the Poisson likelihood should be more symmetric.

In the topright graph of Figure 1 we have two intervals for λ.

<r code>

```
poisson_lkl <- function(lambda, data, fn) {
    n <- length(data)
    par <- fn(lambda)
    # log-likelihood ignoring irrelevant constant terms
    lkl = -n * par + sum(data) * log(par)
    return(lkl)
}
poisson_fisherinfo <- length(data_1)/mean(data_1)

taylor2ndorder <- function(lkl,
                           fisher_type = c("analytic", "numeric"),
                           fisher_exp = NULL,
                           data, par, par_est, fn) {
    fisherinfo <- switch(fisher_type,
                         "analytic" = fisher_exp,
                         "numeric" = hessian(lkl, x = par_est, data, fn))
    lkl(par_est, data, fn) - .5 * fisherinfo * (par - par_est)^2
}
cut_lkl <- log(.15) + poisson_lkl(mean(data_1), data_1, fn = I)
cut_taylor <- log(.15) + taylor2ndorder(poisson_lkl,
                                        fisher_type = "analytic",
                                        fisher_exp = poisson_fisherinfo,
                                        data = data_1,
                                        par = mean(data_1),
                                        par_est = mean(data_1), fn = I)
cut <- cut_lkl <- cut_taylor

zero_function <- function(lkl, par, cut, ...) {
    lkl(par, ...) - cut
}
lambda_zeros <- rootSolve::uniroot.all(zero_function,
                                       range(seq(.25, 2, .1)),
                                       lkl = poisson_lkl,
                                       data = data_1, fn = I, cut = cut)
wald <- function(par_est, se) { par_est + qnorm(c(.025, .975)) * se }
lambda_wald <- wald(par_est = mean(data_1),
                    se = sqrt(1/poisson_fisherinfo))
wes_poisson <- wes_palette("Royal1", 2, "continuous")
```

```r
poisson_mainplot <-
    function(seq, mle, fn, zeros, wald, lab_x, lab_y, end_y) {
        ggplot() +
            geom_vline(xintercept = mle, linetype = "dotted") +
            stat_function(data = data.frame(lambda = seq),
                          fun = poisson_lkl,
                          args = list(data = data_1, fn),
                          aes(x = lambda, color = wes_poisson[1])) +
            stat_function(data = data.frame(lambda = seq),
                          fun = taylor2ndorder,
                          args = list(lkl = poisson_lkl,
                                      fisher_type = "analytic",
                                      fisher_exp = poisson_fisherinfo,
                                      data = data_1, par_est = mle, fn),
                          aes(x = lambda, color = wes_poisson[2])) +
            theme_minimal() +
            labs(x = lab_x, y = lab_y,
                 subtitle =
                     "In dashed, cutoff point corresponding to a 95% CI,",
                 caption = "In dotted, the MLE.") +
            theme(plot.subtitle = element_text(size = 12),
                  plot.caption = element_text(size = 11),
                  legend.text = element_text(size = 12)) +
            scale_colour_manual(NULL, values = wes_poisson,
                                labels =
                                    c("log-like", "quadratic approx.")) +
            geom_hline(yintercept = cut, linetype = "dashed") +
            geom_segment(data = data.frame(
                             x = c(zeros, wald), y = rep(cut, 4),
                             xend = c(zeros, wald), yend = rep(end_y, 4)),
                         aes(x = x, y = y, xend = xend, yend = yend),
                         color = rep(wes_poisson, each = 2),
                         arrow = arrow(length = unit(.03, "npc")))
}
poisson_zoomplot <- function(seq, mle, fn, lab_x, lab_y) {
    ggplot(data.frame(lambda = seq), aes(x = lambda)) +
        geom_vline(xintercept = mle, linetype = "dotted") +
        stat_function(fun = poisson_lkl,
                      args = list(data = data_1, fn),
                      color = wes_poisson[1]) +
        stat_function(fun = taylor2ndorder,
                      args = list(lkl = poisson_lkl,
                                  fisher_type = "analytic",
                                  fisher_exp = poisson_fisherinfo,
                                  data = data_1, par_est = mle, fn),
                      color = wes_poisson[2]) +
        theme_minimal() +
```

```
        labs(x = lab_x, y = lab_y)
}
grid.arrange(poisson_mainplot(seq = seq(.25, 2, .1),
                              mle = mean(data_1), fn = I,
                              zeros = lambda_zeros, wald = lambda_wald,
                              lab_x = expression(lambda),
                              lab_y = TeX("$l(\\lambda; y)$"),
                              end_y = -13) +
             labs(tag = "A"),
             poisson_zoomplot(seq = seq(.8, 1.2, .1),
                              mle = mean(data_1), fn = I,
                              lab_x = expression(lambda),
                              lab_y = TeX("$l(\\lambda; y)$")) +
             labs(tag = "B"), heights = c(1, 1/3), ncol = 1)
```

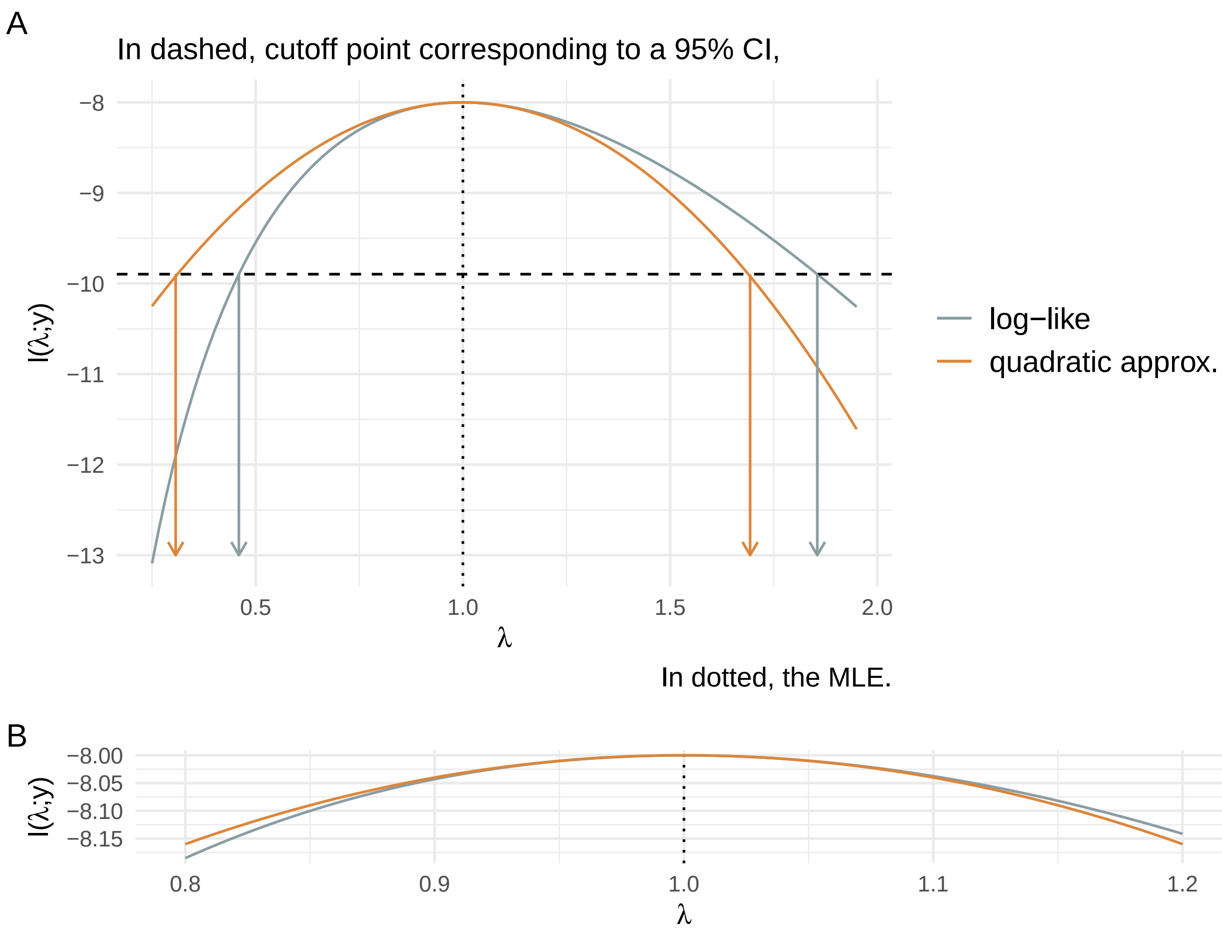

Figure 1: log-likelihood function and MLE of λ in a Poisson(λ). A: quadratic approximation and two intervals for λ, one based in the likelihood and one based in the quadratic approximation. B: a better look in the quadratic approximation.

In Figure 1A we have a interval based in the likelihood, $\lambda \in (0.459, 1.855)$, but with a cutoff criterion based in a χ^2 distribution. Thus, to have a nominal 95% confidence interval (CI) we use a cutoff $c = 15\%$. This interval is based in a large-sample theory, and since we're dealing here with a reasonable regular case, this interval shows as a good approximation.

Still in Figure 1A, we have a interval based in the quadratic approximation of the log-likelihood, $\lambda \in (0.307, 1.693)$. From the quadratic approximation we get

$$\log \frac{L(\lambda)}{L(\hat{\lambda})} \approx -\frac{1}{2} I_O(\hat{\lambda})(\lambda - \hat{\lambda})^2,$$

that also follows a χ^2 distribution, since we have a r.v. $\hat{\lambda}$ normalized (expected value subtracted and divided by its variance) and squared. From this we get the following exact (in the normal case) 95% confidence interval

$$\hat{\lambda} \pm 1.96 \; I_O(\hat{\lambda})^{-1/2} \qquad (\, \hat{\lambda} \pm 1.96 \; \text{se}(\hat{\lambda}) \,).$$

In the nonnormal cases this is an approximate 95% CI.

The actual variance is $I_E(\hat{\lambda})^{-1/2}$, but for the Poisson case the Fisher (expected) information is equal to the observed one, $I_O(\hat{\lambda})^{-1/2}$.

A very nice thing that we can see from this intervals is that a Wald interval (a $\pm$ interval) corresponds to a cut in the quadratic approximation exactly in the same point that the probability-based interval cuts the (log-)likelihood. Thus, as more regular the likelihood, better will be the fit of the approximation and more reliable will be the Wald interval.

(b) Repita a questão anterior para a reparametrização $\theta = \log \lambda$.

By the invariance property of the MLE we get

$$\hat{\lambda} = \bar{x} \quad \Rightarrow \quad g(\hat{\lambda}) = \log \hat{\lambda} = \hat{\theta} = \log \bar{x} = g(\bar{x}).$$

i.e., the MLE of $\hat{\theta}$ is $\log \bar{x}$.

By the Delta Method we compute the variance of $\hat{\theta}$,

$$V[\theta] = V[g(\lambda)] = \left[\frac{\partial}{\partial \lambda} g(\lambda) \right]^2 V[\lambda] = \left[\frac{1}{\lambda} \right]^2 \frac{\lambda}{n} = \frac{1}{\lambda \, n} \quad \rightarrow \quad V[\hat{\theta}] = \frac{1}{\bar{x} \, n}.$$

From this we can take the observed information for the reparametrization

$$V[\hat{\theta}] = I_O^{-1}(\hat{\theta}) = (\bar{x} \, n)^{-1}.$$

Now we do, in the same manner, everything that we did in the previous letter.

<r code>

```
theta_zeros <- rootSolve::uniroot.all(zero_function, range(seq(-1, 1, .1)),
                                      lkl = poisson_lkl,
                                      data = data_1, fn = exp, cut = cut)
poisson_fisherinfo <- sum(data_1)
```

```
theta_wald <- wald(par_est = log(mean(data_1)),
                   se = sqrt(1/poisson_fisherinfo))
grid.arrange(
    poisson_mainplot(seq = seq(-1, 1, .1),
                     mle = log(mean(data_1)), fn = exp,
                     zeros = theta_zeros, wald = theta_wald,
                     lab_x = expression(theta),
                     lab_y = TeX("$l(\\theta; y)$"), end_y = -14) +
    labs(tag = "A"),
    poisson_zoomplot(seq = seq(- .5, .5, .1),
                     mle = log(mean(data_1)), fn = exp,
                     lab_x = expression(theta),
                     lab_y = TeX("$l(\\theta; y)$")) +
    labs(tag = "B"), heights = c(1, 1/3), ncol = 1)
```

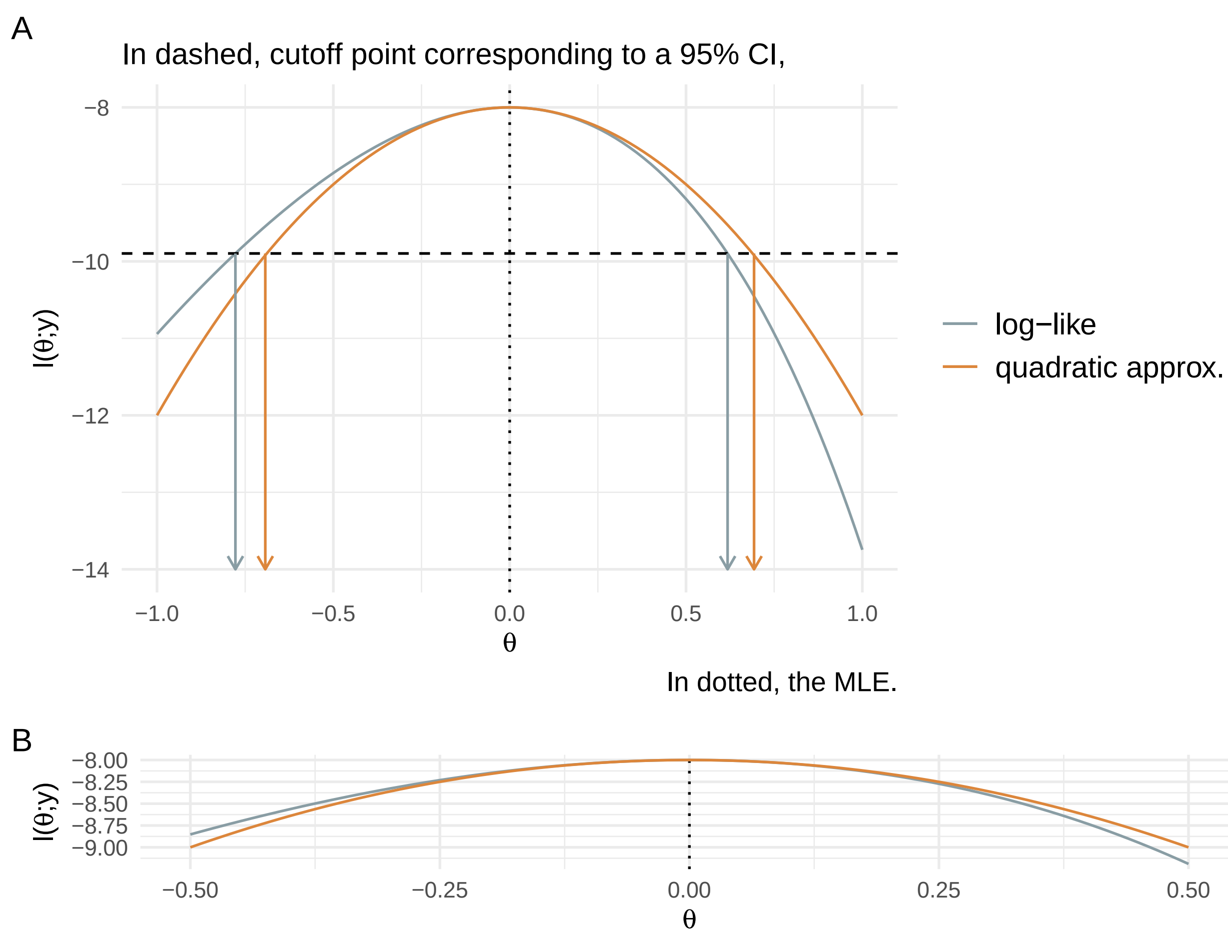

Figure 2: log-likelihood function and MLE of θ in a Poisson($\lambda = e^{\theta}$). A: quadratic approximation and two intervals for θ, one based in the likelihood and one based in the quadratic approximation. B: a better look in the quadratic approximation.

We obtained here two intervals for θ. One based in a cut in the likelihood, $\theta \in (-0.778, 0.618)$ - in Figure 2A, and one based in a cut in the quadratic approximation of the likelihood, $\theta \in (-0.693, 0.693)$ - also in Figure 2A.

With the parametrization $\theta = \log \lambda$ the two intervals are closer than the ones obtained for λ. i.e., for θ (with the use of the log) we get a more regular likelihood.

(c) Obtenha ainda (por pelo menos dois métodos diferentes) intervalos de confiança para o parâmetro λ a partir da função de verossimilhança (aproximada ou não) de θ.

<r code>

```
poisson_mainplot(seq = seq(0, 2, .1),
                 mle = log(mean(data_1)), fn = exp,
                 zeros = exp(theta_zeros), wald = exp(theta_wald),
                 lab_x = expression(theta),
                 lab_y = TeX("$l(\\theta; y)$"),
                 end_y = -45) +
    geom_vline(xintercept = mean(data_1), linetype = "dotted")
```

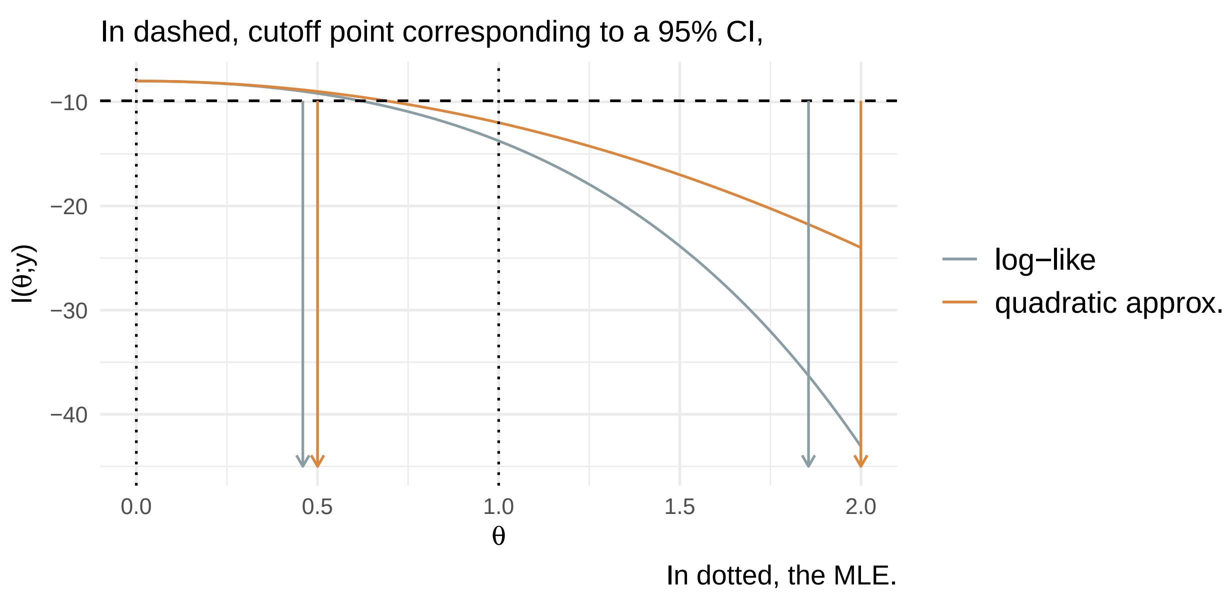

Figure 3: log-likelihood function and quadratic aprroximation of θ, and MLE of λ in a Poisson($\lambda = e^{\theta}$). The confidence intervals are for λ.

Since the MLE has the invariance property, a very simple idea is: take the obtained interval for θ and apply a transformation, $\lambda = e^{\theta}$. This simple idea is true for the interval obtained via likelihood function, as we can see in Figure 3. The obtained interval is the same that the one in letter (a). However, this idea doesn't work for the interval based in the quadratic approximation.

In letter (a), via the quadratic approximation we got $\lambda \in (0.307, 1.693)$. Here, applying the relation $\lambda = e^\theta$ we get $\lambda \in (0.5, 2)$. i.e., this shows that the invariance property applies to the likelihood function itself, not to its quadratic approximation.

Exercise 2: dist. Binomial

Seja X uma v.a. com distribuição binomial com $n = 12$. Obtenha a função de verossimilhança para cada uma das observações a seguir e desenhe todas em um mesmo gráfico, escalonado se necessário.

$$X \sim \text{Binomial}(n = 12, \theta) \equiv \mathbb{P}[X = x \mid n = 12, \theta] = \binom{n}{x} \theta^x (1-\theta)^{n-x}.$$

(a) $x = 5$,

$$L(\theta) = \mathbb{P}[X = 5 \mid n = 12, \theta] = \binom{12}{5} \theta^5 (1-\theta)^{12-5} = 792\, \theta^5 (1-\theta)^7.$$

(b) $x \leq 10$,

$$L(\theta) = \mathbb{P}[X \leq 10 \mid n = 12, \theta] = \sum_{x=0}^{10} \binom{12}{x} \theta^x (1-\theta)^{12-x}.$$

(c) $3 \leq x \leq 7$,

$$L(\theta) = \mathbb{P}[3 \leq X \leq 7 \mid n = 12, \theta] = \sum_{x=3}^{7} \binom{12}{x} \theta^x (1-\theta)^{12-x}.$$

<r code>

```r
binom <- function(theta, x) {
    sum(sapply(x, function(i) dbinom(x = i, size = 12, prob = theta)))
}
binom_vec <- Vectorize(binom, "theta")

fish_binom <- fish(3, option = "Trimma_lantana")

ggplot(data.frame(theta = seq(0, 1, .1)), aes(x = theta)) +
    stat_function(fun = binom_vec, args = list(x = 5),
                  size = 1, aes(color = fish_binom[1])) +
    stat_function(fun = binom_vec, args = list(x = seq(0, 10, .1)),
                  size = 1, aes(color = fish_binom[2])) +
    stat_function(fun = binom_vec, args = list(x = seq(3, 7, .1)),
                  size = 1, aes(color = fish_binom[3])) +
    theme_minimal() +
    labs(title = "Likelihoods",
         x = expression(theta), y = TeX("$l(\\theta; x)$")) +
    theme(legend.text = element_text(size = 12)) +
    scale_colour_manual(NULL, values = fish_binom,
                        labels = c("(a)", "(b)", "(c)"))
```

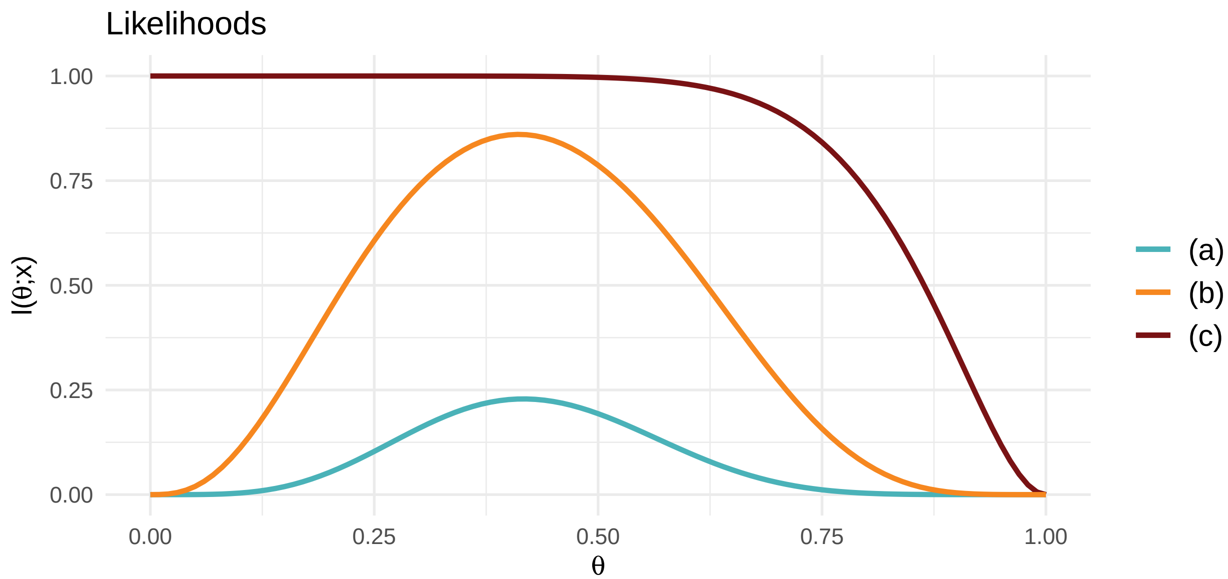

Figure 4: Likelihood functions for different sets of observations.

Exercise 3: dist. Hipergeométrica

A fim de se obter uma estimativa do público de um jogo sem utilizar dados de venda de ingressos ou registros das roletas do estádio, foram distribuídas camisas especiais para 300 torcedores sob a condição que estes a utilizassem durante um jogo. Durante o jogo foram selecionados ao acaso 250 torcedores verificando-se que 12 destes possuiam a camisa.

(a) Obtenha a função de verossimilhança para o número total de torcedores.

We have here a random variable, let's say, X, representing the number of observed successes, k. Here an observed success is select a fan using a special shirt. So, we have

$$X \sim \text{Hypergeometric}(N, K = 300, n = 250),$$

with probability $p = K/N$.

N is the population size, that we want estimate. K is the unknown number of fans using a special shirt, and n is the number of, randomly, selected fans.

$$\begin{aligned}\Pr[X = 12] = \frac{\binom{300}{12}\binom{N-300}{250-12}}{\binom{N}{250}} &= \frac{300!\ 250!}{288!\ 238!\ 12!}\frac{(N-300)!\ (N-250)!}{(N-538)!\ N!}\\ &= \text{constant} \times \frac{(N-300)!\ (N-250)!}{(N-538)!\ N!}.\end{aligned}$$

Since we already have a k, the likelihood is equal to $\Pr[X = k]$.

$$L(N) = \Pr[X = 12] = \text{constant}\frac{(N-300)!\,(N-250)!}{(N-538)!\,N!}$$
$$l(N) = \log L(N) = \log\ \text{constant} + \log\ \frac{(N-300)!\,(N-250)!}{(N-538)!\,N!}$$
$$\approx \log\ \frac{(N-300)!\,(N-250)!}{(N-538)!\,N!}.$$

The MLE here is given by $\hat{N} = \lfloor Kn/k \rfloor = 6250$. A plot of the likelihood is provided in Figure 5, and it present some very interesting behaviours.

In Figure 5A, going from $N = 3150$ to $N = 16000$, we clearly see the likelihood asymmetry. In B, we focus around the MLE to better see the curvature. Around that point we have a very small variation, 0.3, for a range of $3000N$'s. This shows how smooth is the curvature around the MLE in a large region.

<r code>

```r
hypergeo_lkl <- function(N, n, K, k) {
    l1 <- lgamma(N - K + 1)
    l2 <- lgamma(N - n + 1)
    l3 <- lgamma(N - K - n + k + 1)
    l4 <- lgamma(N + 1)
    lkl <- l1 + l2 - l3 - l4
    return(lkl)
}
hypergeo_plot <- function(seq_N, n, K, k) {
    ggplot() +
        stat_function(data = data.frame(N = seq_N),
                      aes(x = N),
                      fun = hypergeo_lkl,
                      args = list(n = n, K = K, k = k)) +
        geom_vline(xintercept = 6250, linetype = "dotted") +
        theme_minimal() +
        labs(y = TeX("$l(N; K, n)$"), caption = "In dotted, the MLE.")
}
grid.arrange(
    hypergeo_plot(seq(3150, 16000, 100),
                  n = 250, K = 300, k = 12) + labs(tag = "A"),
    hypergeo_plot(seq(5000, 8000, 100),
                  n = 250, K = 300, k = 12) + labs(tag = "B"),
    ncol = 2)
```

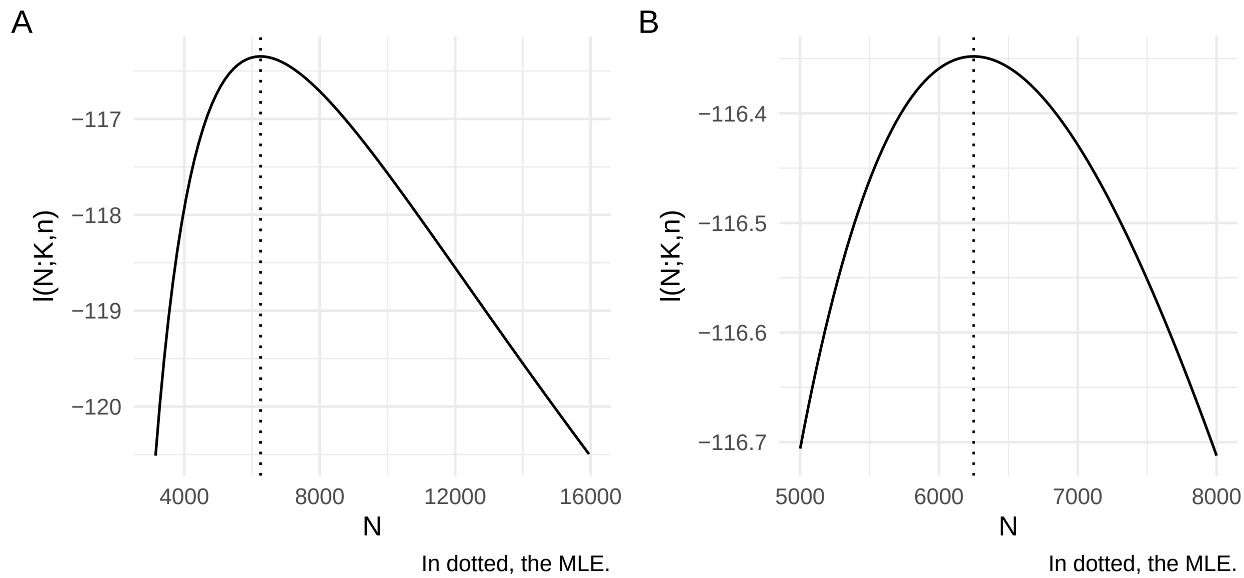

Figure 5: log-likelihood of N in a Hypergeometric(N, K = 300, n = 250) based on a $k = 12$. A and B: different ranges of N to have a better understanding of the likelihood behaviour.

(b) Obtenha a estimativa pontual e a intervalar, esta última por pelo menos dois métodos diferentes.

<r code>

```
optimize(hypergeo_lkl, n = 250, K = 300, k = 12,
         lower = 539, upper = 2e4, maximum = TRUE)
```

```
$maximum
[1] 6249.502

$objective
[1] -116.3481
```

Using the `optimize` routine to do the optimization, we obtain $\hat{N} = 6249.5$ and a log-likelihood of -116.35. This value for $\hat{N}$ is practically equal to what the ML theory says, $\hat{N} = \lfloor Kn/k \rfloor = 6250$.

Next, we compute a quadratic approximation for the hypergeometric (log-)likelihood followed by two intervals, one based in the likelihood, and a wald interval based in the quadratic approximation. All this is presented in Figure 6.

<r code>

```
hypergeoATmle <- hypergeo_lkl(6250, n = 250, K = 300, k = 12)
cut <- log(.15) + hypergeoATmle
```

```r
# this taylor2ndorder approx. is extremely general, however, because of
# the "..." its a mess when we put it into the stat_function
taylor2ndorder <-
    function(lkl,
             fisher_type = c("analytic", "numeric"), fisher_exp = NULL,
             par, par_est, ...) {
        fisherinfo <- switch(fisher_type,
                             "analytic" = fisher_exp,
                             "numeric" = hessian(lkl, x = par_est, ...))
        lkl(par_est, ...) - .5 * fisherinfo * (par - par_est)^2
    }
taylorATmle <- taylor2ndorder(hypergeo_lkl, fisher_type = "numeric",
                              par = 6250, par_est = 6250,
                              n = 250, K = 300, k = 12)
## cut <- log(.15) + taylorATmle

# I need here a new zero_function, since the hypergeo_lkl has an
# argument called "n", as the uniroot.all... this detail == Error
zero_function <- function(lkl, par, cut, ...) {
    lkl(par, n = 250, ...) - cut
}
hypergeo_zeros <-
    rootSolve::uniroot.all(zero_function, range(seq(3000, 12000, 100)),
                           lkl = hypergeo_lkl, K = 300, k = 12, cut = cut)
hypergeo_fisherinfo <-
    (-1) * as.vector(hessian(hypergeo_lkl, 6250, n = 250, K = 300, k = 12))

hypergeo_wald <- wald(par_est = 6250, se = sqrt(1/hypergeo_fisherinfo))

hypergeo_plots <- function(seq_N, n, K, k) {
    hypergeo_plot(seq_N, n, K, k) +
        stat_function(aes(x = seq_N),
                      fun = function(x) {
                          hypergeoATmle -
                              .5 * hypergeo_fisherinfo * (x - 6250)^2
                      }, linetype = "dashed")
}
grid.arrange(
    hypergeo_plots(seq(3000, 11000, 100), n = 250, K = 300, k = 12) +
    geom_hline(yintercept = cut, linetype = "dotted") +
    geom_segment(aes(x = c(hypergeo_zeros, hypergeo_wald),
                     y = rep(cut, 4),
                     xend = c(hypergeo_zeros, hypergeo_wald),
                     yend = rep(-122, 4)),
                 linetype = rep(c("solid", "dashed"), each = 2),
                 arrow = arrow(length = unit(.03, "npc"))) +
    labs(title = "In solid, likelihood; In dashed, its quadratic approx.",
```

```
    tag = "A"),
hypergeo_plots(seq(5750, 6750, 100), n = 250, K = 300, k = 12) +
labs(caption = NULL, tag = "B"), heights = c(1, 1/3), ncol = 1)
```

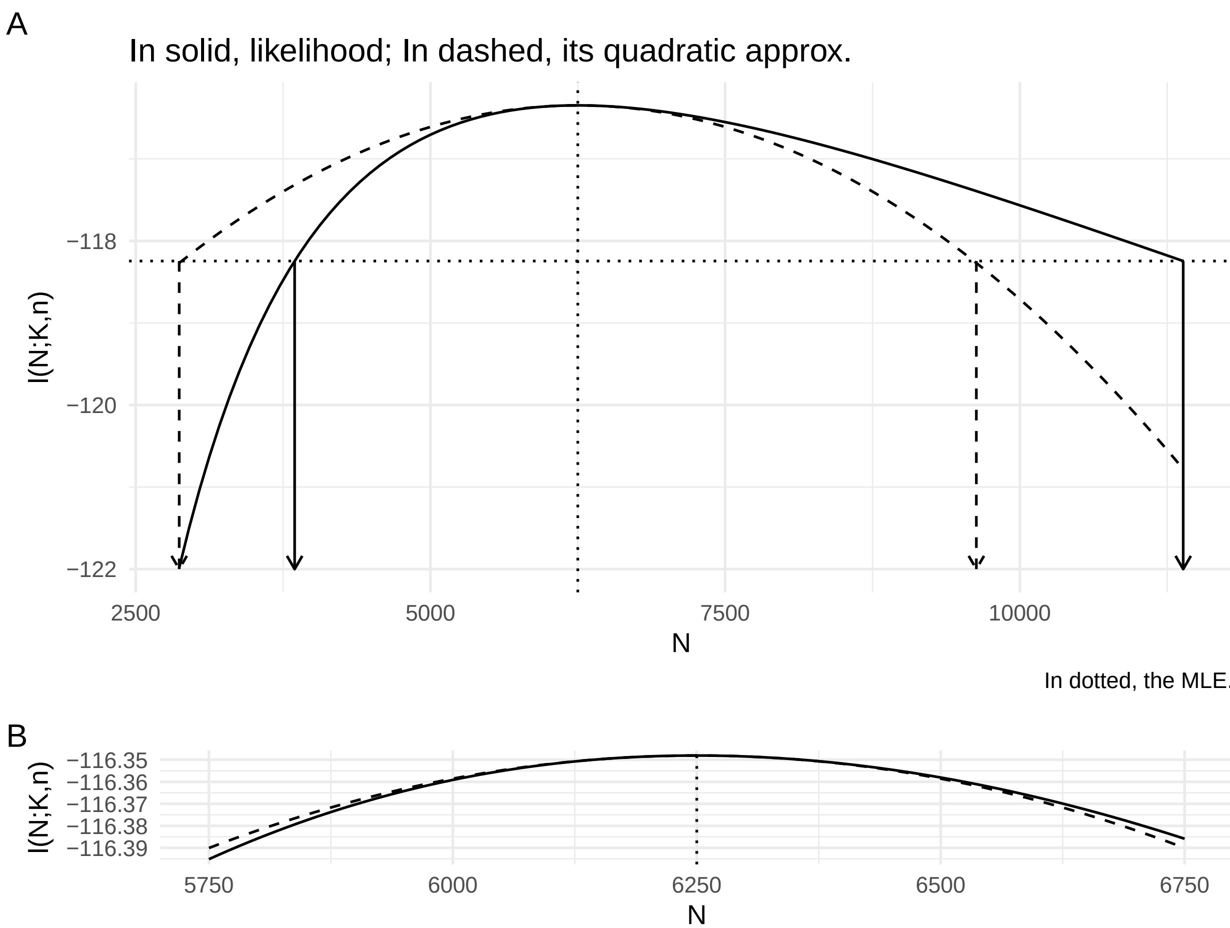

Figure 6: log-likelihood function and MLE of N in a Hypergeometric(N, K = 300, n = 250) based on a $k = 12$. A: a quadratic approximation and two intervals for N, onebased in the likelihood and one based in the quadratic approximation. B: a better look in the quadratic approximation.

The quadratic approximation and the intervals were obtained in the same way that in the exercises before, with the only difference that here the hessian was obtained numerically, just for simplicity.

Based in a cut in the likelihood, we obtain $(3848, 11386)$ as a 95% interval for the total number of fans in the game. With the wald interval (quadratic approximation) we obtain $(2869, 9631)$ as a 95% interval. *A considerable difference*.

(c) Repita e compare os resultados caso fossem 500 camisas e 20 com camisas dentre os 250.

The procedure here is exactly the same, we have

$$\Pr[X = 20] = \frac{\binom{500}{20}\binom{N-500}{250-20}}{\binom{N}{250}} = \text{constant} \times \frac{(N-500)!\,(N-250)!}{(N-730)!\,N!}.$$

With likelihood

$$\begin{aligned} L(N) &= \Pr[X = 20] = \text{constant}\frac{(N-500)!\,(N-250)!}{(N-730)!\,N!} \\ l(N) = \log L(N) &= \log\ \text{constant} + \log\ \frac{(N-500)!\,(N-250)!}{(N-730)!\,N!} \\ &\approx \log\ \frac{(N-500)!\,(N-250)!}{(N-730)!\,N!}. \end{aligned}$$

In Figure 7 we provide the likelihood graphs and intervals for N. The MLE is exactly the same that the one in the previous scenario, since

$$\hat{N} = \lfloor Kn/k \rfloor = \lfloor 300 \times 250\ /\ 12 \rfloor = \lfloor 500 \times 250\ /\ 20 \rfloor = 6250.$$

<r code>

```
hypergeoATmle <- hypergeo_lkl(6250, n = 250, K = 500, k = 20)
cut <- log(.15) + hypergeoATmle

taylorATmle <- taylor2ndorder(hypergeo_lkl, fisher_type = "numeric",
                              par = 6250, par_est = 6250,
                              n = 250, K = 500, k = 20)
## cut <- log(.15) + taylorATmle
hypergeo_zeros <-
    rootSolve::uniroot.all(zero_function, range(seq(4000, 10000, 100)),
                           lkl = hypergeo_lkl, K = 500, k = 20, cut = cut)
hypergeo_fisherinfo <-
    (-1) * as.vector(hessian(hypergeo_lkl, 6250, n = 250, K = 500, k = 20))

hypergeo_wald <- wald(par_est = 6250, se = sqrt(1/hypergeo_fisherinfo))

grid.arrange(
    hypergeo_plots(seq(4000, 10000, 100), n = 250, K = 500, k = 20) +
    geom_hline(yintercept = cut, linetype = "dotted") +
    geom_segment(aes(x = c(hypergeo_zeros, hypergeo_wald),
                     y = rep(cut, 4),
                     xend = c(hypergeo_zeros, hypergeo_wald),
                     yend = rep(-202, 4)),
                 linetype = rep(c("solid", "dashed"), each = 2),
```

```
                arrow = arrow(length = unit(.03, "npc"))) +
labs(title = "In solid, likelihood; In dashed, its quadratic approx.",
     tag = "A"),
hypergeo_plots(seq(5675, 6900, 100), n = 250, K = 500, k = 20) +
labs(caption = NULL, tag = "B"), heights = c(1, 1/3), ncol = 1)
```

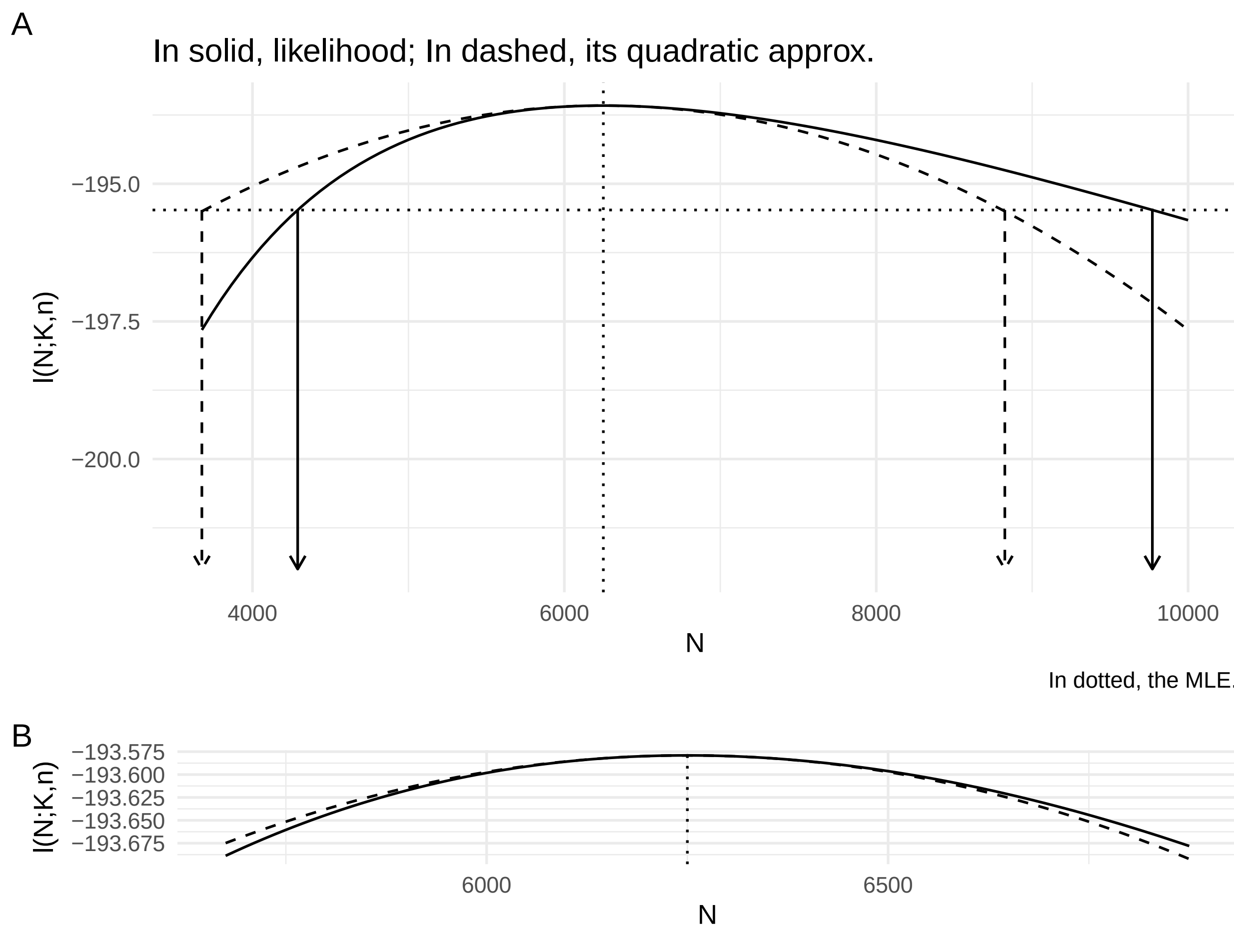

Figure 7: log-likelihood function and MLE of N in a Hypergeometric(N, K = 500, n = 250) based on a $k = 20$. A: a quadratic approximation and two intervals for N, one based in the likelihood and one based in the quadratic approximation. B: a better look in the quadratic approximation.

- Based in a cut in the likelihood, we obtain $(4290, 9771)$ as a 95% interval for the total number of fans in the game.
- With the wald interval (quadratic approximation) we obtain $(3675, 8825)$ as a 95% interval for the total number of fans.

A considerable difference.

Comparing with the previous scenario, the intervals changed considerably - even with MLE retain the same.

The likelihood shape is the same that before, just with a vertical increase. Before the maximum log-likelihood estimate was around -116, now is around -194.

Exercise 4: Estatísticas de ordem Gaussianas

Sejam os dados a seguir provenientes de uma amostra aleatória de $X \sim \text{Normal}(\mu, \sigma^2)$, onde vamos assumir que σ^2 é conhecido e com valor igual a da variância amostral. Considere as seguintes observações

$$73 \quad 75 \quad 84 \quad 76 \quad 93 \quad 79 \quad 85 \quad 80 \quad 76 \quad 78 \quad 80.$$

<r code>

```
data_4 <- c(73, 75, 84, 76, 93, 79, 85, 80, 76, 78, 80)
```

Denote os valores ordenados por $x_{(1)}, x_{(2)}, x_{(3)}, \ldots, x_{(11)}$.

<r code>

```
(ordered_4 <- sort(data_4))

 [1] 73 75 76 76 78 79 80 80 84 85 93

gaussian_lkl <- function(mu, data, weight,
                         data_type = c("easy", "hard"), something) {
    lkl <- switch(data_type,
                  "easy" = sum(sapply(
                      data,
                      function(x) dnorm(x, mean = mu, sd = sd(data_4),
                                        log = TRUE)
                  )),
                  "hard" = something(mu))
    lkl <- weight * lkl
    return(lkl)
}
gaussian_vec <- Vectorize(gaussian_lkl, "mu")
```

Obtenha e compare os gráficos da função de verossimilhança de μ para os seguintes casos:

- (a) O conjunto completo dos dados $x_1, x_2, x_3, \ldots, x_{11}$ está disponível;

$$L(\mu; x) = \prod_{i=1}^{n} \phi\left(\frac{x_i - \mu}{\sigma}\right).$$

- (b) Apenas a média amostral é fornecida;

$$L(\mu; x) = \phi\left(\frac{\bar{x} - \mu}{\sigma}\right)^n.$$

- (c) Apenas a mediana $x_{(6)}$ é fornecida;

$$L(\mu; x) = \Phi\left(\frac{x_{(6)} - \mu}{\sigma}\right)^5 \times \left(1 - \Phi\left(\frac{x_{(6)} - \mu}{\sigma}\right)\right)^5 \times \phi\left(\frac{x_{(6)} - \mu}{\sigma}\right).$$

- (d) Apenas os valores mínimo $x_{(1)}$ e máximo $x_{(n)}$ são fornecidos;

$$L(\mu; x) = n \times \left(1 - \Phi\left(\frac{x_{(1)} - \mu}{\sigma}\right)\right)^{n-1} \times \phi\left(\frac{x_{(1)} - \mu}{\sigma}\right) + n \times \Phi\left(\frac{x_{(n)} - \mu}{\sigma}\right)^{n-1} \times \phi\left(\frac{x_{(n)} - \mu}{\sigma}\right).$$

- (e) Os quartis $(Q_1,\ Q_2,\ Q_3)$ são fornecidos;

$$L(\mu; x) = \Phi\left(\frac{Q_1 - \mu}{\sigma}\right)^{\#\{x<Q_1\}} \times \Phi\left(\frac{Q_2 - \mu}{\sigma}\right)^{\#\{Q_1<x<Q_2\}} \times \left(\frac{Q_3 - \mu}{\sigma}\right)^{\#\{Q_2<x<Q_3\}} \times \left(1 - \Phi\left(\frac{Q_3 - \mu}{\sigma}\right)\right)^{\#\{x>Q_3\}}.$$

- (f) Apenas os dois menores valores $x_{(1)}$ e $x_{(2)}$ são fornecidos.

$$\begin{aligned} L(\mu; x) &= L(\mu; x_{(1)}) + L(\mu; x_{(2)}) \\ &= n\phi\left(\frac{x_{(1)} - \mu}{\sigma}\right)\binom{n-1}{1-1}\Phi\left(\frac{x_{(1)} - \mu}{\sigma}\right)^{1-1}\left(1 - \Phi\left(\frac{x_{(1)} - \mu}{\sigma}\right)\right)^{n-1} \\ &+ n\phi\left(\frac{x_{(2)} - \mu}{\sigma}\right)\binom{n-1}{2-1}\Phi\left(\frac{x_{(2)} - \mu}{\sigma}\right)^{2-1}\left(1 - \Phi\left(\frac{x_{(2)} - \mu}{\sigma}\right)\right)^{n-2}. \end{aligned}$$

<r code>

```
fish_gaussian <- fish(6, option = "Trimma_lantana")

gaussian_plot <- function(mu_grid, data, weight,
                          data_type, something, mle, index, tag) {
    ggplot() +
        stat_function(aes(x = mu_grid),
                      fun = gaussian_vec,
                      args = list(data, weight, data_type, something),
                      color = fish_gaussian[index], size = 1) +
        geom_vline(xintercept = mle, linetype = "dotted") +
        theme_minimal() +
        labs(x = expression(mu), y = TeX("$l(\\mu; x)$"), tag = tag)
}
grid.arrange(
    ## (a) -------------------------------------------------------------
```

```r
gaussian_plot(seq(60, 100, 1), data = data_4, weight = 1,
              data_type = "easy", something = NULL,
              mle = mean(data_4), index = 1, tag = "(a)"),
## (b) ------------------------------------------------------------
gaussian_plot(seq(60, 100, 1), data = mean(data_4), weight = 11,
              data_type = "easy", something = NULL,
              mle = mean(data_4), index = 2, tag = "(b)"),
## (c) ------------------------------------------------------------
gaussian_plot(seq(60, 100, 1), data = NULL, weight = 1,
              data_type = "hard",
              something = function(mu) {
                  core <- pnorm(ordered_4[6], mean = mu, sd(data_4))
                  lkl <- 5 * log(core) + 5 * log(1 - core) +
                      dnorm(ordered_4[6], mu, sd(data_4), log = TRUE)
                  return(lkl)
              }, mle = ordered_4[6], index = 3, tag = "(c)"),
## (d) ------------------------------------------------------------
gaussian_plot(seq(60, 105, 1), data = NULL, weight = 1,
              data_type = "hard",
              something = function(mu) {
                  dcore_max <-
                      dnorm(ordered_4[11], mu, sd(data_4), log = T)
                  dcore_min <-
                      dnorm(ordered_4[1], mu, sd(data_4), log = T)
                  pcore_max <- pnorm(ordered_4[11], mu, sd(data_4))
                  pcore_min <- pnorm(ordered_4[1], mu, sd(data_4))
                  lkl <- 2 * log(11) +
                      10 * log(pcore_max * (1 - pcore_min)) +
                      dcore_max + dcore_min
                  return(lkl)
              }, mle = mean(c(ordered_4[1], ordered_4[11])),
              index = 4, tag = "(d)"),
## (e) ------------------------------------------------------------
gaussian_plot(seq(55, 100, 1), data = NULL, weight = 1,
              data_type = "hard",
              something = function(mu) {
                  n1 <- sum(data_4 < quantile(data_4)[[2]])
                  Q1 <- pnorm(quantile(data_4)[[2]], mu, sd(data_4))
                  n2 <- sum(data_4 < quantile(data_4)[[3]] &
                            data_4 > quantile(data_4)[[2]])
                  Q2 <- pnorm(quantile(data_4)[[3]], mu, sd(data_4))
                  n3 <- sum(data_4 < quantile(data_4)[[4]] &
                            data_4 > quantile(data_4)[[3]])
                  Q3 <- pnorm(quantile(data_4)[[4]], mu, sd(data_4))
                  n4 <- sum(data_4 > quantile(data_4)[[4]])
                  lkl <- n1 * log(Q1) + n2 * log(Q2) + n3 * log(Q3) +
                      n4 * log(1 - Q3)
```

```
                return(lkl)
            }, mle = mean(quantile(data_4)[2:4]),
            index = 5, tag = "(e)"),
## (f) ------------------------------------------------------------
gaussian_plot(seq(65, 100, 1), data = NULL, weight = 1,
            data_type = "hard",
            something = function(mu) {
                lkl_piece <- function(mu, index) {
                    F <- pnorm(ordered_4[index], mu, sd(data_4))
                    lkl <- log(11) +
                        dnorm(ordered_4[index], mu, sd(data_4),
                              log = TRUE) +
                        log(choose(10, index - 1)) +
                        (index - 1) * log(F) +
                        (11 - index) * log(1 - F)
                    return(lkl)
                }
                lkl_1 <- lkl_piece(mu, index = 1)
                lkl_2 <- lkl_piece(mu, index = 2)
                lkl <- lkl_1 + lkl_2
                return(lkl)
            }, mle = NULL, index = 6, tag = "(f)"),
nrow = 2, ncol = 3)
```

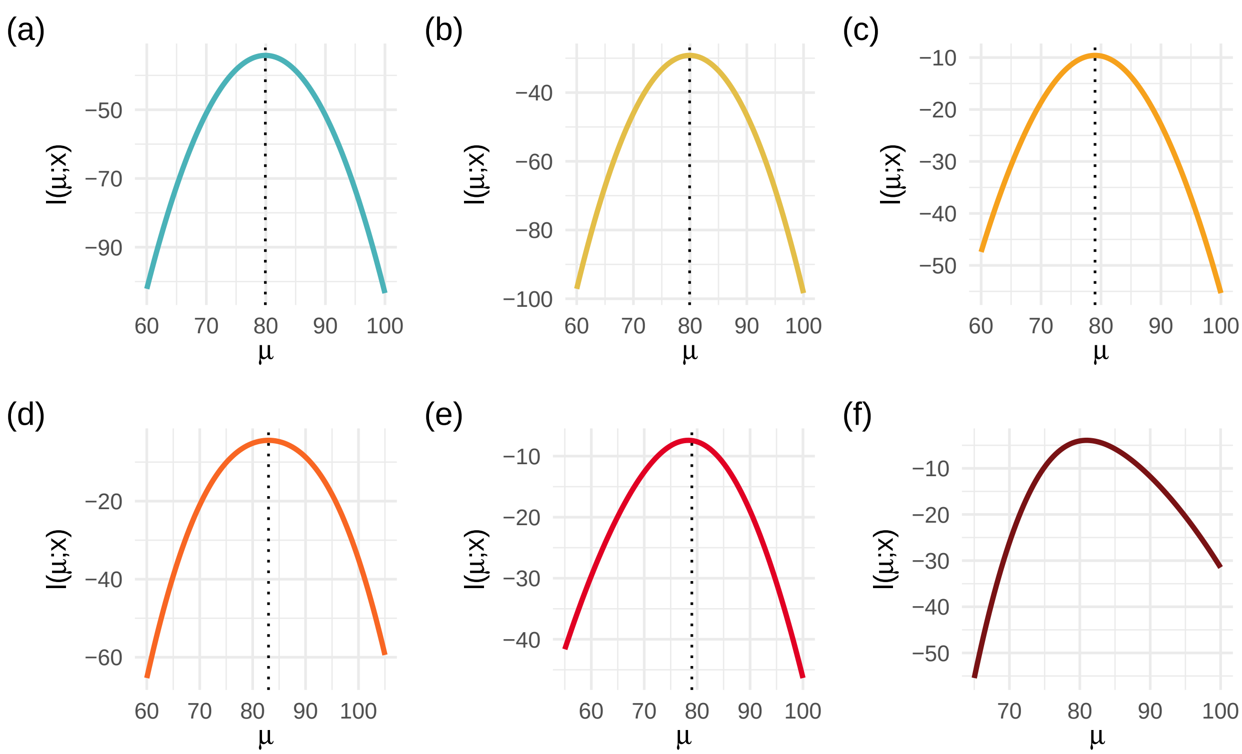

Figure 8: Likelihood functions for μ. In dotted, the MLEs.

Exercise 5: dist. Gaussiana

O rendimento, X_i, de um campo i de trigo é considerado como sendo normalmente distribuído com média θz_i onde z_i é uma quantidade conhecida de fertilizante aplicado no campo. Para um novo campo a quantidade de fertilizante pode ser escolhida, mas o rendimento é aleatório para uma dada quantidade de fertilizante. Presumindo-se que os rendimentos em diferentes campos são independentes (mas não identicamente distribuídos como mudanças no rendimento com a quantidade de fertilizante usada) uns dos outros. Deseja-se estimar o rendimento para uma proporção de fertilizante, isto é, a estimativa de θ. Especificamente, $X_1, \ldots, X_n$ são variáveis aleatórias independentes com distribuição

$$X_i \sim \text{Normal}(\theta Z_i, 1)$$

para $i = 1, \ldots, n$, onde $z_1, \ldots, z_n$ são conhecidos (possivelmente diferentes) constantes positivas.

- (a) Encontre $\hat{\theta}$.

$$\begin{aligned} l(\theta; X) &= -\frac{n}{2}\log 2\pi - n\log 1 - \frac{1}{2}\|X - Z\theta\|^2 \\ &= -\frac{n}{2}\log 2\pi - \frac{1}{2}\|X - Z\theta\|^2 \\ \frac{\partial l(\theta; X)}{\partial \theta} &= Z^\top (X - Z\theta) \\ &\equiv 0 \\ &\Rightarrow Z^\top Z\hat{\theta} = Z^\top X \\ &\Rightarrow \hat{\theta} = (Z^\top Z)^{-1} Z^\top X. \end{aligned}$$

- (b) Mostre que $\hat{\theta}$ é um estimador não viesado, isto é, $\mathbb{E}\hat{\theta} = \theta$ (lembre-se que os valores de z_i são constantes). Verifique suas respostas pegando $z_i = 1$ para $i = 1, \ldots, n$.

$$\mathbb{E}\hat{\theta} = \mathbb{E}[(Z^\top Z)^{-1} Z^\top X] = (Z^\top Z)^{-1} Z^\top \mathbb{E}X = (Z^\top Z)^{-1} Z^\top Z\theta = \theta.$$

- (c) Suponha que o rendimento tenha a seguinte propriedade, $X_1, \ldots, X_n$ são variáveis aleatórias independentes com distribuição

$$X_i \sim \text{Normal}(\theta Z_i, Z_i^2)$$

para $i = 1, \ldots, n$, onde $z_1, \ldots, z_n$ são constantes conhecidas positivas. Encontre $\hat{\theta}_l$, $\hat{\theta}$

e $\hat{\theta}_u$ com $c^* = 3.84$.

$$\begin{aligned}
l(\theta; X) &= -\frac{n}{2} \log 2\pi - n \log Z^\top Z - \frac{1}{2IZ^\top Z} \|X - Z\theta\|^2 \\
&= -\frac{n}{2} \log 2\pi - -\frac{1}{2}(Z^\top Z)^{-1}\frac{1}{2} \|X - Z\theta\|^2 \\
\frac{\partial l(\theta; X)}{\partial \theta} &= (Z^\top Z)^{-1} Z^\top (X - Z\theta) \\
&\equiv 0 \\
&\Rightarrow (Z^\top Z)^{-1} Z^\top Z\hat{\theta} = (Z^\top Z)^{-1} Z^\top X \\
&\Rightarrow \hat{\theta} = (Z^\top Z)^{-1} Z^\top X.
\end{aligned}$$

$c^* = 3.84$ consiste em obter um intervalo de confiança com poder nominal de 95%.

Pensando num intervalo baseado num corte da função de verossimilhança, caímos num problema que consiste simplesmente em encontrar os zeros de uma função, i.e., precisamos de valores para X.

Outra possibilidade é pensar num intervalo do tipo Wald, baseado na aproximação quadrática da função de verossimilhança ao redor de $\hat{\theta}$. Como $X \sim$ Normal, é de se esperar que tal aproximação seja boa.

$c^* = 3.84$ é baseado numa distribuição χ^2_1 e equivale a um corte em 15% da função de verossimilhança/aproximação quadrática. Tal valor é equivalente ao conhecido valor de 1.96 da Normal padrão.

De tal maneira, $\hat{\theta}_l$ e $\hat{\theta}_u$ são respectivamente,

$$\hat{\theta} \pm 1.96 \text{ se}[\hat{\theta}].$$

Com

$$\begin{aligned}
\text{se}[\theta] = I_E^{-1/2}\,[\theta] = \mathbb{E}^{-1/2}\, I_O[\theta] &= -\mathbb{E}^{-1/2}\, H[\theta] \\
&= -\mathbb{E}^{-1/2}\, \frac{\partial^2 l(\theta; X)}{\partial \theta^2} \\
&= -\mathbb{E}^{-1/2}\, \frac{\partial}{\partial \theta}(Z^\top Z)^{-1} Z^\top (X - Z\theta) \\
&= \mathbb{E}^{-1/2}\, (Z^\top Z)^{1} Z^\top Z \\
&= \mathbb{E}^{-1/2}\, I \\
&= I.
\end{aligned}$$

Assim,

$$\hat{\theta}_l = \hat{\theta} - 1.96, \quad \text{e} \quad \hat{\theta}_u = \hat{\theta} + 1.96.$$

- (d) Explique em palavras por que $\hat{\theta}$ é diferente dos encontrados anteriormente.

  Em termos da estimativa pontual, $\hat{\theta}$, o fato da variância de X ser fixa e conhecida, ou dependente de Z, não mostrou fazer diferença. Agora, esta diferença teve impacto na hora de calcular $\hat{\theta}_l$ e $\hat{\theta}_u$, já que no segundo cenário obtivemos uma matriz identidade.

Exercise 6: dist. Exponencial

Considere um modelo exponencial de parâmetro θ. Encontre um intervalo de confiança para θ por pelo menos dois métodos e faça um estudo de simulação para verificar a taxa de cobertura de cada método.

$$L(\theta; x) = \prod_{i=1}^{n} \theta e^{-\theta x_i} \quad \Rightarrow \quad \log L(\theta; x) = l(\theta; x) = n \log \theta - \theta \sum_{i=1}^{n} x_i = n \log \theta - \theta \, n\bar{x}.$$

Quadratic approximation (2nd order Taylor expansion) around the MLE, $\hat{\theta} = \bar{x}^{-1}$:

$$l(\theta; x) = l(\hat{\theta}; x) - \frac{1}{2}(\theta - \hat{\theta})^2 \, I_O(\hat{\theta}) = l(\hat{\theta}; x) - \frac{1}{2}(\theta - \hat{\theta})^2 \, \frac{n}{\hat{\theta}^2} = l(\hat{\theta}; x) - \frac{n}{2}\left(\frac{\theta - \hat{\theta}}{\hat{\theta}}\right)^2.$$

<r code>

```
set.seed(131019)
data_6 <- rexp(n = 100, rate = 5)

exp_lkl <- function(theta, data) {
    n <- length(data)
    lkl <- n * log(theta) -  theta * n * mean(data)
    return(lkl)
}
exp_taylor <- function(theta, data) {
    theta_hat <- 1/mean(data)
    const <- length(data)/2
    taylor <- exp_lkl(theta_hat, data) -
        const * ((theta - theta_hat)/theta_hat)^2
    return(taylor)
}
cut <- log(.15) + exp_lkl(1/mean(data_6), data_6)
## log(.15) + exp_taylor(1/mean(data_6), data_6)
theta_seq <- seq(3.5, 6.5, .1)

find_zero <- function(theta) {
    exp_lkl(theta, data_6) - cut
}
theta_lkl <- rootSolve::uniroot.all(find_zero, range(theta_seq))

exp_wald <- function(theta) {
    n <- length(data_6)
    theta + qnorm(c(.025, .975)) * theta/sqrt(n)
}
theta_wald <- exp_wald(1/mean(data_6))

ggplot() +
```

```
stat_function(aes(theta_seq),
              fun = exp_lkl, args = list(data = data_6)) +
stat_function(aes(theta_seq),
              fun = exp_taylor, args = list(data = data_6),
              linetype = "dashed") +
geom_vline(xintercept = 1/mean(data_6), linetype = "dotted") +
geom_hline(yintercept = cut, linetype = "dotted") +
geom_segment(aes(x = c(theta_lkl, theta_wald), y = rep(cut, 4),
                 xend = c(theta_lkl, theta_wald), yend = rep(54, 4)),
             linetype = rep(c("solid", "dashed"), each = 2),
             arrow = arrow(length = unit(.03, "npc"))) +
theme_minimal() +
labs(x = expression(theta), y = TeX("$l(\\theta; x)$"),
     title = "Likelihood, in solid. Quadratic approx., in dashed.",
     subtitle = "The MLE, in x-dotted. Function cut, in y-dotted.")
```

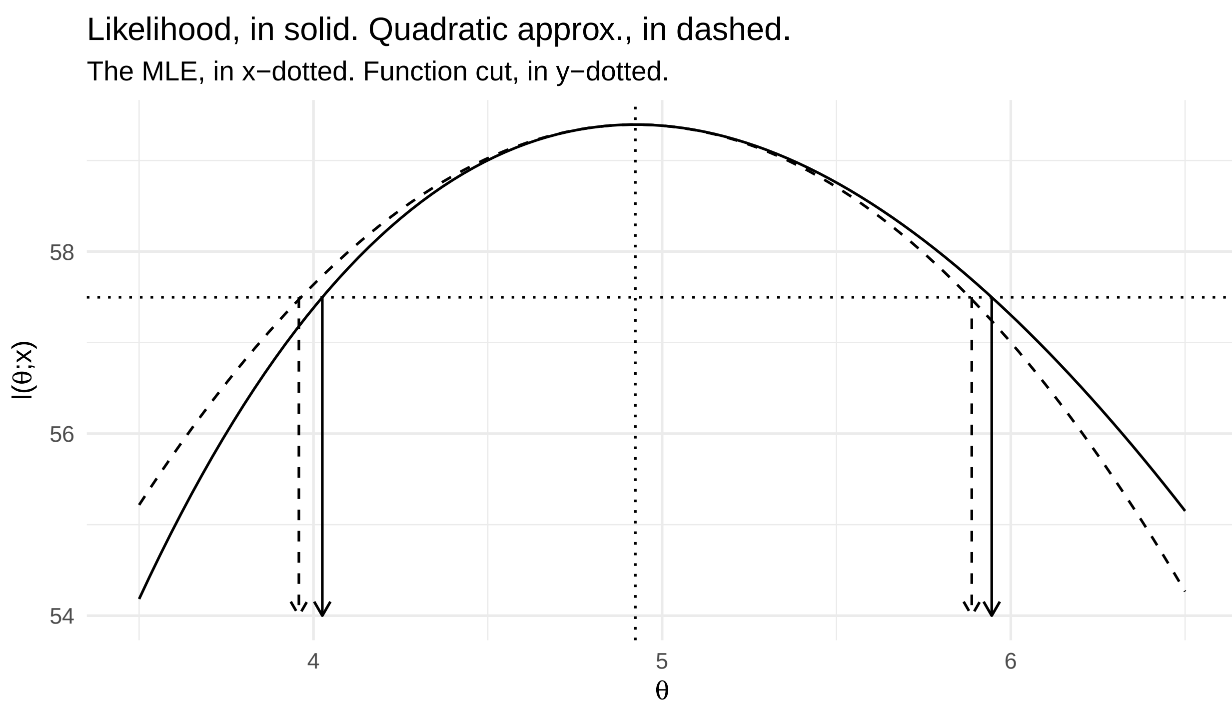

Figure 9: log-likelihood function (in solid), and MLE of θ in a Exponential(θ) based on a sample of size 100. Quadratic approximation around the MLE in dashed line.

Agora, com o intuito de verificar a taxa de cobertura dos intervalos exibidos na Figura 9, simulamos 1000 amostras de tamanho 100 de uma Exponencial de parâmetro $\theta = 5$.

<r code>

```
fish_exp <- fish(2, option = "Hypsypops_rubicundus")
```

```
replications <- replicate(1e3, 1/mean(rexp(n = 100, rate = 5)))
ggplot() +
    geom_histogram(aes(replications), binwidth = .25,
                   col = fish_exp[1], fill = fish_exp[2], alpha = .5) +
    geom_vline(xintercept = mean(replications), linetype = "dashed") +
    labs(x = expression(theta), y = "Frequency") +
    theme_minimal()
```

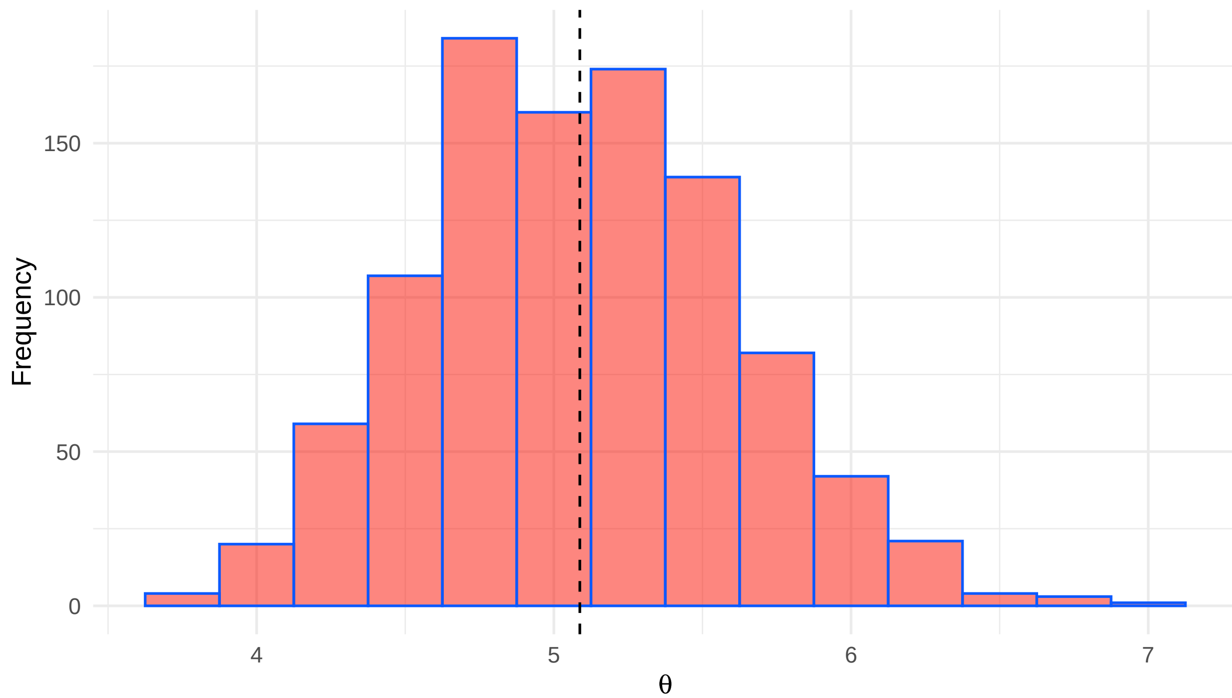

Figure 10: Histogram made of 1000 simulations. In dashed, the average.

```
sum(replications >= theta_lkl[1] & replications <= theta_lkl[2])
```

```
[1] 930
```

```
sum(replications >= theta_wald[1] & replications <= theta_wald[2])
```

```
[1] 925
```

Como os intervalos obtidos possuem poder nominal de 95%, se espera que em torno de 950 das 1000 médias amostrais estejam contidas nos intervalos. Os valores obtidos são próximos de tal valor, com uma taxa de cobertura um pouco inferior para o intervalo obtido via aproximação quadrática, o que é de se esperar aqui por dois motivos: 1) como se observa no histograma da Figura 10, não temos uma distribuição "muito Normal", i.e, o histograma apresenta uma visível assimetria; 2) se trata de um intervalo baseado na aproximação da função de verossimilhança, não na verossimilhança em si.

Exercise 7: reparametrização da dist. Exponencial

Considere o modelo do exercício anterior e uma reparametrização onde $\lambda = 1/\theta$. Encontre o intervalo de confiança para λ por pelo menos três métodos e avalia a taxa de cobertura de cada um via simulação. Qual método você considera o melhor? Justifique.

<r code>

```
exp_lkl <- function(lambda, data) {
    n <- length(data)
    lkl <- -n * log(lambda) - n * mean(data)/lambda
    return(lkl)
}
exp_taylor <- function(lambda, data) {
    lambda_hat <- mean(data)
    const <- length(data)/2
    taylor <- exp_lkl(lambda_hat, data) -
        const * ((lambda - lambda_hat)/lambda_hat)^2
    return(taylor)
}
cut <- log(.15) + exp_lkl(mean(data_6), data_6)
## log(.15) + exp_taylor(mean(data_6), data_6)
## theta_seq <- seq(3.5, 6.5, .1)
lambda_seq <- seq(1/6.5, 1/4, length.out = 50)
## interval based on a cut in the likelihood ---------------------------
lambda_lkl <- rootSolve::uniroot.all(find_zero, range(lambda_seq))
## interval based on a cut in the quadratic approximation --------------
lambda_wald <- exp_wald(mean(data_6))
## interval based on the application of the delta method ---------------
lambda_delta <- mean(data_6) +
    qnorm(c(.025, .975)) * mean(data_6)/sqrt(length(data_6))

lambda_wald == lambda_delta
```

```
[1] TRUE TRUE
```

A princípio podemos pensar em quatro intervalos para λ: 1) intervalo baseado em sua própria verossimilhança, o que é equivalente a fazer $1/\text{CI}_\theta$, onde CI_θ é o intervalo obtido para θ; 2) baseado em sua aproximação quadrática; 3) baseado no intervalo obtido via aproximação quadrática para θ; e 4) via método Delta.

O intervalo 3) é o pior, dado que ele é baseado na aproximação quadrática de uma diferente função (agora estamos interessados em λ, não mais em θ).

Como mostrado no código anterior, o intervalo obtido via aproximação quadrãtica da verossimilhança de λ é exatamente igual ao obtido via o método Delta, dado por

$$\left\{1/\hat{\theta} - z_{\alpha/2} \left|(\hat{\theta}^{-1})'\right| I_E^{-1/2}(\hat{\theta}) \quad , \quad 1/\hat{\theta} + z_{\alpha/2} \left|(\hat{\theta}^{-1})'\right| I_E^{-1/2}(\hat{\theta})\right\}$$

Assim, na Figura 11 temos dois intervalos, o baseado no corte da verossimilhança, e o baseado em sua aproximação quadrática.

<r code>

```
ggplot() +
    stat_function(aes(lambda_seq),
                  fun = exp_lkl, args = list(data = data_6)) +
    stat_function(aes(lambda_seq),
                  fun = exp_taylor, args = list(data = data_6),
                  linetype = "dashed") +
    geom_vline(xintercept = mean(data_6), linetype = "dotted") +
    geom_hline(yintercept = cut, linetype = "dotted") +
    geom_segment(aes(x = c(lambda_lkl, lambda_wald), y = rep(cut, 4),
                     xend = c(lambda_lkl, lambda_wald), yend = rep(55, 4)),
                 linetype = rep(c("solid", "dashed"), each = 2),
                 arrow = arrow(length = unit(.03, "npc"))) +
    theme_minimal() +
    labs(x = expression(lambda), y = TeX("$l(\\lambda; x)$"),
         title = "Likelihood, in solid. Quadratic approx., in dashed.",
         subtitle = "The MLE, in x-dotted. Function cut, in y-dotted.")
```

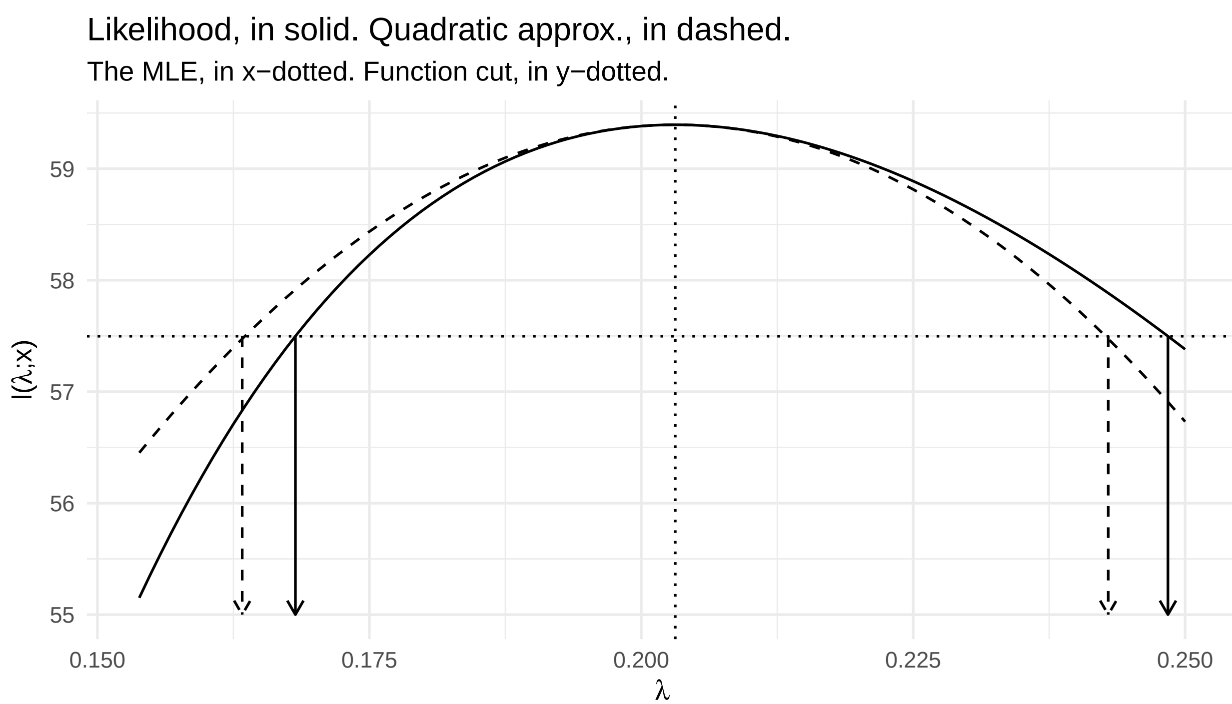

Figure 11: log-likelihood function (in solid), and MLE of λ in a Exponential($\lambda = 1/\theta$) based on a sample of size 100. Quadratic approximation around the MLE in dashed line.

Para verificar a taxa de cobertura dos intervalos exibidos na Figura 11, simulamos 1000 amostras de tamanho 100 de uma Exponencial de parâmetro $\theta = 5$.

<r code>

```
replications <- replicate(1e3, mean(rexp(n = 100, rate = 5)))
ggplot() +
    geom_histogram(aes(replications), binwidth = .01,
                   col = fish_exp[1], fill = fish_exp[2], alpha = .5) +
    geom_vline(xintercept = mean(replications), linetype = "dashed") +
    labs(x = expression(lambda), y = "Frequency") + theme_minimal()
```

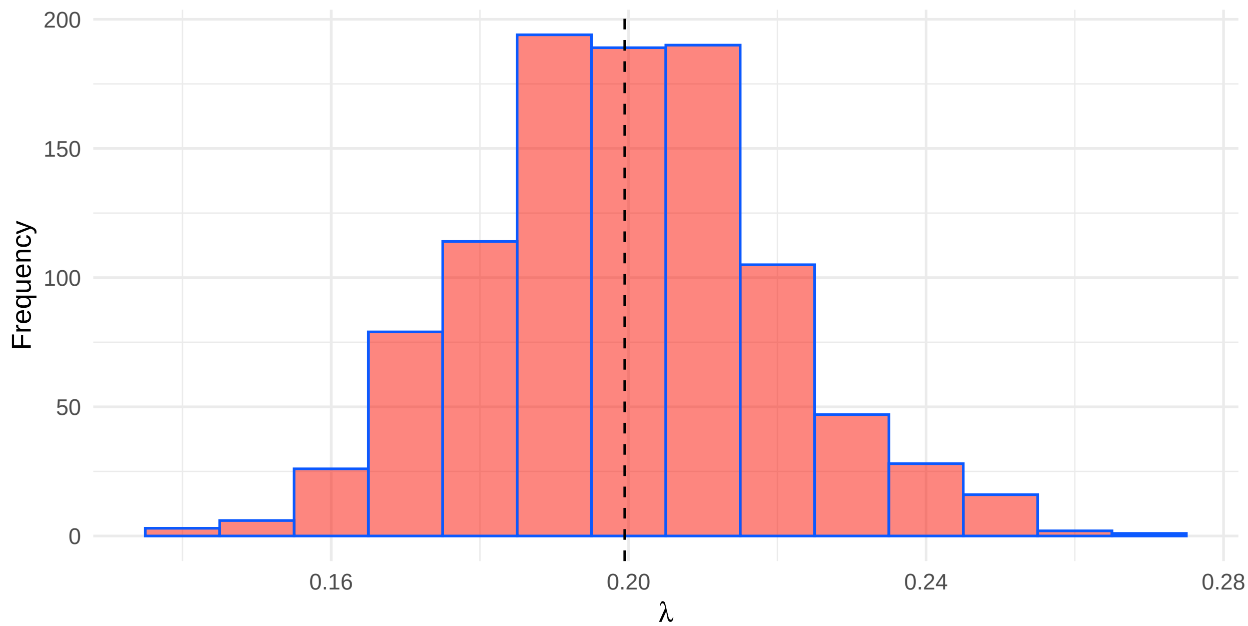

Figure 12: Histogram made of 1000 simulations. In dashed, the average.

```
sum(replications >= lambda_lkl[1] & replications <= lambda_lkl[2])
```

```
[1] 934
```

```
sum(replications >= lambda_wald[1] & replications <= lambda_wald[2])
```

```
[1] 949
```

Como os intervalos obtidos possuem poder nominal de 95%, se espera que em torno de 950 das 1000 médias amostrais estejam contidas nos intervalos. Os valores obtidos são próximos de tal valor, com uma taxa de cobertura um pouco melhor para o intervalo obtido via aproximação quadrática, o que faz sentido, já que aqui temos uma dist. muito mais Normal (Figura 12).

Last modification on ...

```
[1] "2019-10-18 11:12:01 -03"
```