

Lista Prática

Question 1.

1.1. FUNÇÕES DE VEROSSIMILHANÇA E ESTIMADORES

1) A mistura de normais é caracterizada da seguinte forma:

$$(1) \quad e_i \sim \begin{cases} N(\mu_1, \sigma_1^2), \text{ com probabilidade } 1/2, \\ N(\mu_2, \sigma_2^2), \text{ com probabilidade } 1/2. \end{cases}$$

A densidade de $f(e_i)$ é, portanto:

$$(2) \quad f(e_i) = \frac{1}{2} \times \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2} (y_i - \beta_0 - \beta_1 x_i - \mu_1)^2\right) + \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2} (y_i - \beta_0 - \beta_1 x_i - \mu_2)^2\right) \right].$$

Podemos escrever a função de verossimilhança amostral como

$$(3) \quad \begin{aligned} L(\beta|\mathbf{y}, \mathbf{x}; \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) &= \prod_{i=1}^n \frac{1}{2} \times \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2} (y_i - \beta_0 - \beta_1 x_i - \mu_1)^2\right) + \right. \\ &\quad \left. \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2} (y_i - \beta_0 - \beta_1 x_i - \mu_2)^2\right) \right] \\ &= \frac{1}{2^n} \prod_{i=1}^n \times \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2} (y_i - \beta_0 - \beta_1 x_i - \mu_1)^2\right) + \right. \\ &\quad \left. \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2} (y_i - \beta_0 - \beta_1 x_i - \mu_2)^2\right) \right]. \end{aligned}$$

Obtendo a log-verossimilhança:

$$(4) \quad \begin{aligned} l(\beta|\mathbf{y}, \mathbf{x}; \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) &= -n \log(2) + \sum_{i=1}^n \log \left(\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2} (y_i - \beta_0 - \beta_1 x_i - \mu_1)^2\right) + \right. \\ &\quad \left. \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2} (y_i - \beta_0 - \beta_1 x_i - \mu_2)^2\right) \right). \end{aligned}$$

O Estimador de Máxima Verossimilhança é dado por

$$(5) \quad \hat{\beta} = \underset{\beta_0, \beta_1}{\operatorname{argmax}} l(\beta|\mathbf{y}, \mathbf{x}; \mu_1, \mu_2, \sigma_1^2, \sigma_2^2).$$

2) Função de verossimilhança amostral para u , com $u_i \sim \Gamma(a, b)$, definida para $u_i > 0$:

$$(6) \quad \begin{aligned} L(\beta|\mathbf{y}, \mathbf{x}; a, b) &= \prod_{i=1}^n \frac{b^a}{\Gamma(a)} (y_i - \beta_1 - \beta_2 x_i)^{a-1} \exp(-b(y_i - \beta_1 - \beta_2 x_i)) \\ &= \left(\frac{b^a}{\Gamma(a)} \right)^n \prod_{i=1}^n (y_i - \beta_1 - \beta_2 x_i)^{a-1} \exp\left(-b \sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i)\right). \end{aligned}$$

Aplicando transformação logarítmica

$$(7) \quad l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}; a, b) = n \log \left(\frac{b^a}{\Gamma(a)} \right) + (a-1) \sum_{i=1}^n \log(y_i - \beta_1 - \beta_2 x_i) - b \sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i).$$

O Estimador de Máxima Verossimilhança é dado por

$$(8) \quad \hat{\boldsymbol{\beta}} = \underset{\beta_0, \beta_1}{\operatorname{argmax}} l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}; a, b).$$

3) Sabemos que $v_i \sim \text{Cauchy}(0, 1) \sim t(\nu = 1)$, em que ν representa os graus de liberdade. Portanto, a função de verossimilhança amostral para \mathbf{v} , com $v_i \sim \text{Cauchy}(0, 1)$ pode ser escrita como:

$$(9) \quad \begin{aligned} L(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}; \nu) &= \prod_{i=1}^n \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(y_i - \beta_1 - \beta_2 x_i)^2}{\nu}\right) \\ &= \left[\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \right]^n \prod_{i=1}^n \left(1 + \frac{(y_i - \beta_1 - \beta_2 x_i)^2}{\nu}\right). \end{aligned}$$

A log-verossimilhança, por sua vez,

$$(10) \quad l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}; \nu) = n \left[\log \left(\Gamma\left(\frac{\nu+1}{2}\right) \right) - \log(\sqrt{\nu\pi}) - \log \left(\Gamma\left(\frac{\nu}{2}\right) \right) \right] + \sum_{i=1}^n \log \left(1 + \frac{(y_i - \beta_1 - \beta_2 x_i)^2}{\nu} \right).$$

O Estimador de Máxima Verossimilhança é dado por

$$(11) \quad \hat{\boldsymbol{\beta}} = \underset{\beta_0, \beta_1}{\operatorname{argmax}} l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}; \nu).$$

1.2. DISTRIBUIÇÕES ASSINTÓTICAS

Vide [1], Teorema 14.1.M2, *se são satisfeitas condições de regularidade*,

$$(12) \quad \hat{\boldsymbol{\theta}} \xrightarrow{d} N(\boldsymbol{\theta}_0, \{\mathbf{I}(\boldsymbol{\theta}_0)\}^{-1}),$$

em que $\hat{\boldsymbol{\theta}}$ é o estimador de Máxima Verossimilhança de $\boldsymbol{\theta}_0$, e $\mathbf{I}(\boldsymbol{\theta}_0) = -\mathbb{E} \left[\frac{\partial^2 \log(L)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'} \right]$.

Condições de regularidade

- R.1. As primeiras três derivadas de $\log(f(y_i|\boldsymbol{\theta}))$ em relação a $\boldsymbol{\theta}$ são contínuas e finitas para quase todo y_i , e para todo $\boldsymbol{\theta}$.
- R.2. As condições necessárias para obter as esperanças da primeira e da segunda derivada de $\log(f(y_i|\boldsymbol{\theta}))$ são satisfeitas.
- R.3. Para todos os valores de $\boldsymbol{\theta}$, $|\partial^3 \log(f(y_i|\boldsymbol{\theta}))/\partial \theta_j \partial \theta_k \partial \theta_l|$ é menor que uma função com esperança finita.

Sabe-se que uma condição suficiente para que valha R.2 é que o suporte da distribuição não dependa dos parâmetros. Em particular, isto não é satisfeito para o modelo 2.

Question 2.

2.1. SIMULAÇÕES

O experimento de simulação é conduzido em R [3]¹. Primeiramente, definimos parâmetros gerais para a simulação, como o tamanho da amostra e os valores verdadeiros para β .

```
# Simulating data generating processes

n = 2000 # Sample size

# Setting random seed
set.seed(1981)

# Exercise 2
# Generating x
x = rnorm(n = n, mean = 5, sd = sqrt(1))

# Defining betas
beta = c(1.5, 3)
```

Em seguida, simulamos os três processos de geração de dados. Para obter a mistura de normais, é utilizada uma variável auxiliar $P \sim \text{Bernoulli}(1/2)$ e definimos $e_1 \sim N(\mu_1, \sigma_1^2)$, $e_2 \sim N(\mu_2, \sigma_2^2)$ e $e = Pe_1 + (1 - P)e_2$.

```
# Simulating model 1
# Since \mu_1, \mu_2, \sigma_1 and \sigma_2 were not specified,
# I'm considering the following
mu_1 = -2
mu_2 = 2
sigma_1 = sqrt(.5)
sigma_2 = sqrt(25)

e1 = rnorm(n = n, mean = mu_1, sd = sigma_1)
e2 = rnorm(n = n, mean = mu_2, sd = sigma_2)

# Defining an auxiliary random variable P ~ Bernoulli(1/2)

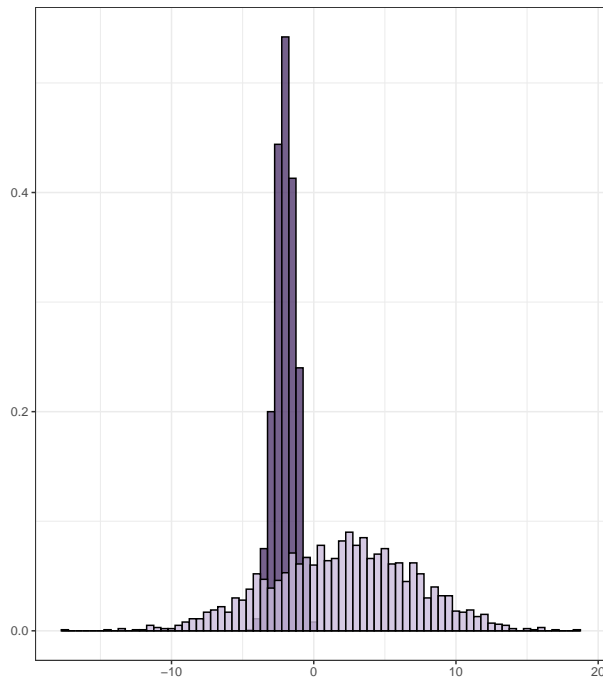
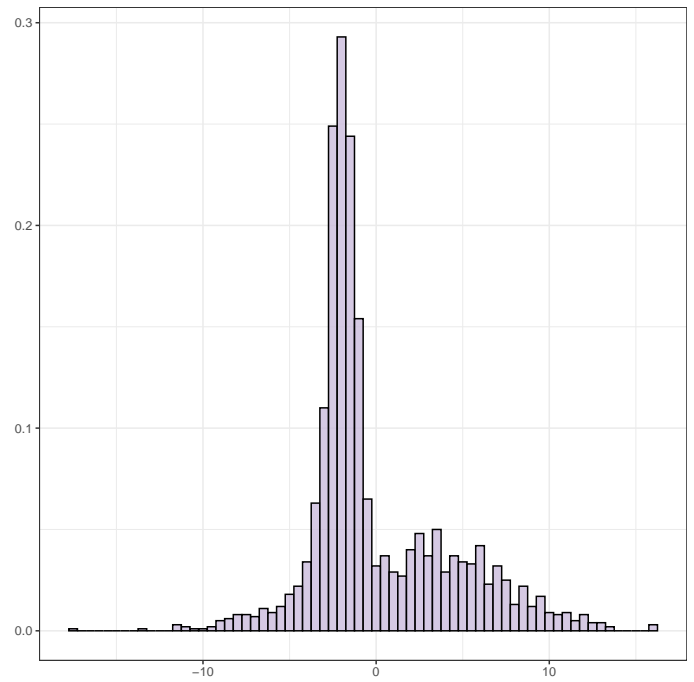
P = rbinom(n = n, size = 1, prob = 1/2)

e = P * e1 + (1-P) * e2

y_1 = beta[1] + beta[2] * x + e
```

```
tibble(e1, e2) %>%
  pivot_longer(cols = c('e1', 'e2')) %>%
  ggplot(aes(x = value, group = name)) +
    geom_histogram(aes(y = after_stat(density), fill = name),
                  alpha = .8, position = 'identity',
                  binwidth = 0.5, color = 'black') +
  labs(fill = '', x = '', y = '') +
```

¹Soluções providas por [5] e [4] foram empregadas na análise e reporte deste exercício.

(A) e_1 e e_2 

(B) Mistura

FIGURA 1. Distribuições de e_1 , e_2 , e

```
scale_fill_manual(values = purple_rain_colors[c(1,3)])

ggplot() +
  geom_histogram(aes(y = after_stat(density), x = e),
                 alpha = .8, position = 'identity',
                 binwidth = 0.5, color = 'black',
                 fill = purple_rain_colors[3]) +
  labs(fill = '', x = '', y = '')
```

```
# Simulating model 2
# Since a and b were not specified, I'm considering the following
```

```
a = 3
b = 6

u = rgamma(n = n, shape = a, rate = b)

y_2 = beta[1] + beta[2] * x + u
```

```
tibble(u) %>%
  ggplot(aes(x = u)) +
    geom_histogram(aes(y = after_stat(density)),
                  alpha = .8, position = 'identity',
                  binwidth = 0.25, color = 'black',
                  fill = purple_rain_colors[3]) +
  labs(fill = '', x = '', y = '')
```

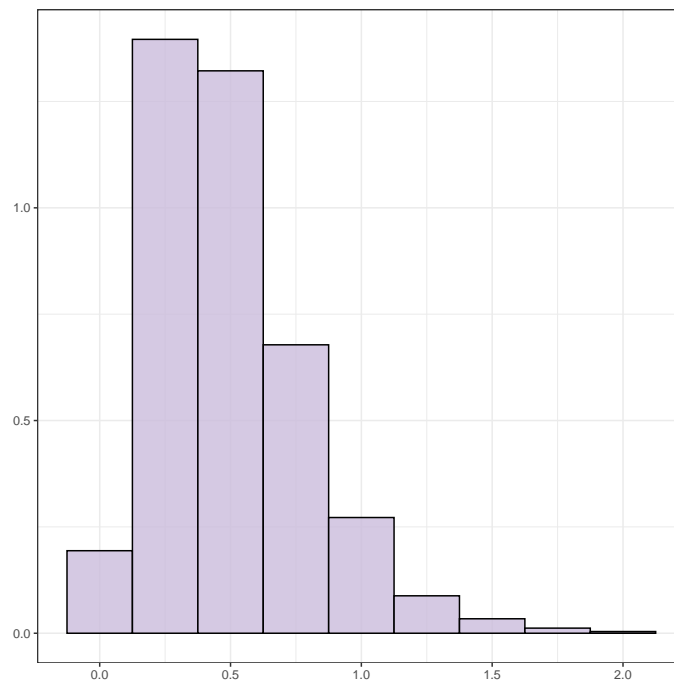


FIGURA 2. Distribuição de u

```
# Simulating model 3
```

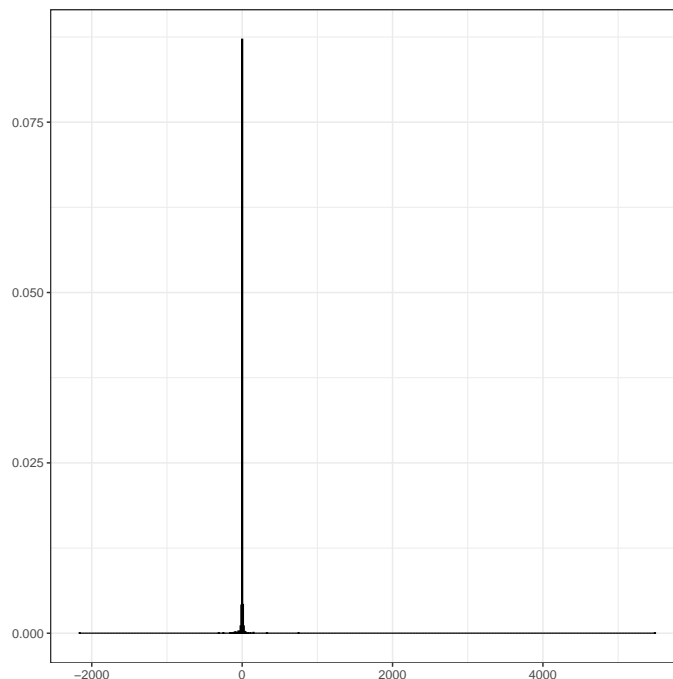
```
v = rt(n = n, df = 1)
```

```
y_3 = beta[1] + beta[2] * x + v
```

```
tibble(v) %>%
  ggplot(aes(x = v)) +
    geom_histogram(aes(y = after_stat(density)),
                   alpha = .8, position = 'identity',
                   binwidth = 10, color = 'black',
                   fill = purple_rain_colors[3]) +
  labs(fill = '', x = '', y = '')
```

```
tibble(y_1, e) %>%
  pivot_longer(cols = c('y_1', 'e')) %>%
  ggplot(aes(x = value, group = name)) +
    geom_histogram(aes(y = after_stat(density), fill = name),
                   alpha = .8, position = 'identity',
                   binwidth = 0.5, color = 'black') +
  labs(fill = '', x = '', y = '') +
  scale_fill_manual(values = purple_rain_colors[c(1,3)])
```

```
tibble(y_2, u) %>%
  pivot_longer(cols = c('y_2', 'u')) %>%
  ggplot(aes(x = value, group = name)) +
    geom_histogram(aes(y = after_stat(density), fill = name),
                   alpha = .8, position = 'identity',
                   binwidth = 0.5, color = 'black') +
```

FIGURA 3. Distribuição de v

```
labs(fill = '', x = '', y = '') +
scale_fill_manual(values = purple_rain_colors[c(1,3)])

tibble(y_3, v) %>%
  pivot_longer(cols = c('y_3', 'v')) %>%
  ggplot(aes(x = value, group = name)) +
    geom_histogram(aes(y = after_stat(density), fill = name),
                  alpha = .8, position = 'identity',
                  binwidth = 0.5, color = 'black') +
  labs(fill = '', x = '', y = '') +
  scale_fill_manual(values = purple_rain_colors[c(1,3)])
```

2.2. ESTATÍSTICAS DESCRITIVAS

```
print(xtable(map_df(list(x, e1, e2, e, y_1, u, y_2, v, y_3),
  ~c(summary(.x), Var. = var(.x))) %>%
  {cbind(data.frame('Variable' = c("$\\bar{x}$", "$\\bar{e}_1$",
    "$\\bar{e}_2$", "$\\bar{e}$",
    "$\\bar{y}_1$", "$\\bar{u}$",
    "$\\bar{y}_2$", "$\\bar{v}$",
    "$\\bar{y}_3$")), .)},
  caption = 'Estatísticas descritivas'),
  sanitize.text.function=function(x){x}, include.rownames = FALSE)
```

Os momentos teóricos, por sua vez, são os seguintes:

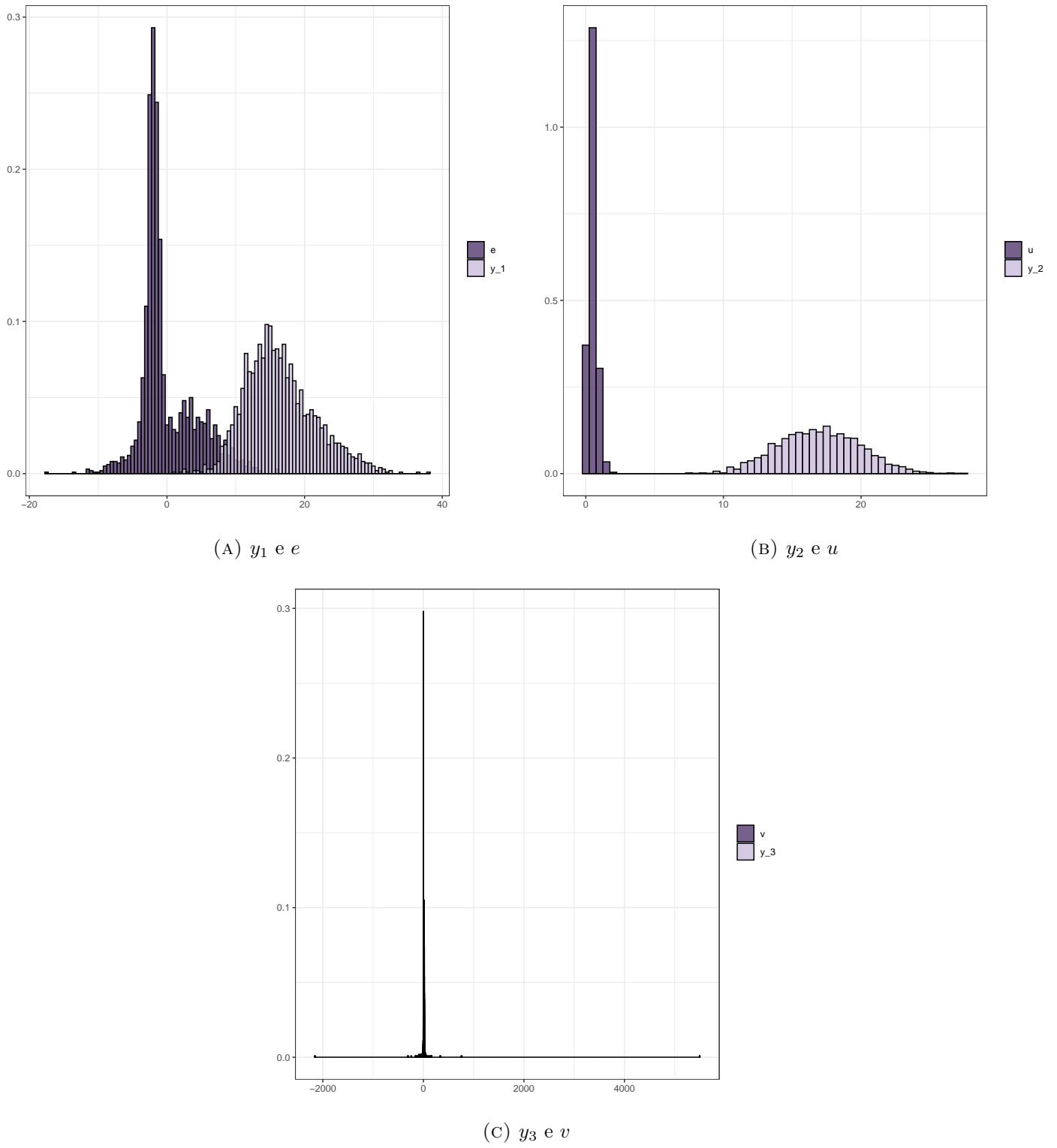


FIGURA 4. Distribuições dos y_i 's e respectivas inovações

$$(13) \quad \mathbb{E}[e_i] = p\mathbb{E}[e_{1i}] + (1-p)\mathbb{E}[e_{2i}] = 0$$

$$(14) \quad \mathbb{V}[e_i] = p(\sigma_1^2 + \mu_1^2) + (1-p)(\sigma_2^2 + \mu_2^2) - [p\mu_1 + (1-p)\mu_2]^2 = 16.75$$

$$(15) \quad \mathbb{E}[y_{1i}] = \beta_0 + \beta_1\mathbb{E}[x_i] + \mathbb{E}[e_i] = 16.5$$

$$(16) \quad \mathbb{V}[y_{1i}] = \mathbb{V}[\beta_1 x_i + e_i] = \beta_1^2 \mathbb{V}[x_i] + \mathbb{V}[e_i] = 25.75$$

Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Var.
\bar{x}	1.81	4.27	4.97	4.98	5.69	8.46	1.00
\bar{e}_1	-4.28	-2.47	-2.00	-2.00	-1.50	0.24	0.51
\bar{e}_2	-17.73	-1.44	2.20	1.97	5.41	18.62	25.27
\bar{e}	-17.73	-2.36	-1.48	-0.02	2.29	16.15	16.76
\bar{y}_1	0.96	12.91	15.77	16.42	19.38	37.87	25.83
\bar{u}	0.02	0.29	0.44	0.50	0.65	1.96	0.08
\bar{y}_2	7.28	14.86	16.91	16.94	19.08	27.26	9.00
\bar{v}	-2163.94	-1.01	-0.00	1.46	0.96	5489.95	17973.98
\bar{y}_3	-2146.88	13.68	16.38	17.91	19.14	5501.13	17953.97

TABELA 1. Estatísticas descritivas

As estimativas observadas para média e variância amostral de y_1 são coerentes com a derivação teórica.

$$(17) \quad \mathbb{E}[u_i] = \frac{a}{b} = 0.5$$

$$(18) \quad \mathbb{V}[u_i] = \frac{a}{b^2} = 0.083333$$

$$(19) \quad \mathbb{E}[y_{2i}] = \beta_0 + \beta_1 \mathbb{E}[x_i] + \mathbb{E}[u_i] = 17$$

$$(20) \quad \mathbb{V}[y_{2i}] = \mathbb{V}[\beta_1 x_i + u_i] = \beta_1^2 \mathbb{V}[x_i] + \mathbb{V}[u_i] = 9.083333$$

As estimativas observadas para média e variância amostral de y_2 não diferem substancialmente dos momentos teóricos.

A esperança e a variância para v_i não estão definidas, portanto, também não estão para y_{3i} . Como sabemos que a distribuição de v_i é simétrica, bem como a de x_i , sugere-se intuitivamente utilizar a mediana como uma medida de centralidade.

Question 3.

- 1) Na ausência de conhecimento sobre o Processo Gerador de Dados, pode-se especular a distribuição dos dados a partir da literatura ou argumentações teóricas sobre a variável em questão, bem como o uso de Análise Exploratória de Dados e comparação de medidas de aderência, sem prejuízo da atenção ao sobreajuste, para diferentes especificações. Alternativamente, pode-se optar por um método que não necessite a especificação de uma distribuição de probabilidade. Note que para a especificação 1 são satisfeitas as hipóteses básicas para o Modelo de Regressão Linear e para a consistência do estimador de Mínimos Quadrados Ordinários.

No entanto, para a segunda especificação, como o suporte da distribuição gama é $(0, \infty)$ e $\mathbb{E}[x_i]$, então seguramente $\mathbb{E}[u_i x_i] \neq 0$. Ou seja, $\hat{\beta}_{OLS} - \beta \not\rightarrow 0$, embora seja convergente. E o modelo 3, por sua vez, quebra as hipóteses do Modelo Regressão Linear e para a consistência do estimador de Mínimos Quadrados Ordinários pois a esperança e a variância de v_i não são definidas.

- 2) Definindo a log-verossimilhança amostral, que será a função objetivo do problema de otimização, como função de β e σ^2 :

```
# Exercise 3.2
# I'm assuming y - Xb ~ N(0, sigma^2)
# For the remaining exercises, I'll be using OLS estimates as initial conditions

# Objective function
l_norm0 <- function(theta, y, x){
  -length(y)/2*log(2*pi*theta[3]) +
    sum(-1/(2*theta[3])*(y - theta[1] - theta[2]*x)^2)
}
```


Iremos utilizar o algoritmo de Nelder-Mead [2] para obter uma solução. Para os três modelos será utilizada como condição inicial a estimativa por Mínimos Quadrados Ordinários.

```
# Model 1

# Initial condition
init_norm = lm(y_1 ~ x) %>%
  {c(coef(.), sigma2 = summary(.)$sigma^2)} # Adding OLS estimate for \sigma^2

mod1_norm <- optim(par = init_norm,
  fn = function(theta){-l_norm0(theta = theta, y = y_1, x=x)})
```

```
print(mod1_norm$par)
## (Intercept)          x      sigma2
##    1.398715    3.015580   16.756951
```

As estimativas para o modelo 1 aparentam ser, ao menos em termos numéricos, coerentes com os valores populacionais dos parâmetros. Inclusive, $\hat{\sigma}^2$ se aproxima bem das variâncias teórica e amostral de e_i .

```
# Model 2

# Initial condition

init_gamma = lm(y_2 ~ x) %>%
  {c(coef(.), sigma2 = summary(.)$sigma^2)}

mod2_norm <- optim(par = init_gamma,
  fn = function(theta){-l_norm0(theta = theta, y = y_2, x=x)})
## Warning in log(2 * pi * theta[3]): NaNs produzidos
```

```
print(mod2_norm$par)
## (Intercept)          x      sigma2
##    2.04442687    2.99021545    0.08333457
```

A estimativa para β_1 se aproxima razoavelmente do parâmetro populacional. A estimativa para σ^2 também é próxima da variância de $u_i \sim \Gamma(a, b)$. No entanto, a estimativa para o intercepto diverge de β_0 .

```
# Model 3

# Initial condition

init_t = lm(y_3 ~ x) %>%
  {c(coef(.), sigma2 = summary(.)$sigma^2)}

mod3_norm <- optim(par = init_t,
  fn = function(theta){-l_norm0(theta = theta, y = y_3, x=x)})
```

```
print(mod3_norm$par)
## (Intercept)          x      sigma2
##    27.139428    -1.853879  17936.131962
```

A estimativa para β difere substancialmente do parâmetro populacional. Notavelmente, o intercepto é numericamente maior, e o parâmetro de inclinação é negativo. Por fim, não é possível comparar $\hat{\sigma}^2$ com $\mathbb{V}[u_i]$ pois esta não é definida.

- 3) Considerando estimação por Máxima Verossimilhança, pode-se considerar conduzir os testes baseados em Máxima Verossimilhança LR, LM e Wald, desde que os modelos satisfaçam as Condições de Regularidade.

Question 4.

Especificando a função objetivo e estimando os parâmetros para o modelo 1:

```
# Exercise 4

# Model 1

# Objective function
l_normmix <- function(theta, mu1, mu2, sigma2_1, sigma2_2, y, x){
  -length(y)*log(2)+ sum(log(1/sqrt(2*pi*sigma2_1)*
    exp(-1/(2*sigma2_1)*
      (y - theta[1] - theta[2] * x - mu1)^2) +
    1/sqrt(2*pi*sigma2_2)*
    exp(-1/(2*sigma2_2) *
      (y - theta[1] - theta[2] * x - mu2)^2)))
}

# Initial condition previously defined #

mod1_true <- optim(par = init_norm[-3],
  fn = function(theta){-l_normmix(theta = theta, mu1 = mu_1,
    mu2 = mu_2, sigma2_1 = sigma_1^2,
    sigma2_2 = sigma_2^2, y = y_1, x=x)})

print(mod1_true$par)
## (Intercept)          x
##    1.684523    2.964243
```

Nota-se que as estimativas, em termos numéricos, não diferem substancialmente das estimativas assumindo distribuição gaussiana. Estas, por sua vez, também são coerentes com os valores populacionais de β .

```
# Model 2

# Objective function
l_gamma <- function(theta, a, b, y, x){
  if(prod(y - theta[1] - theta[2]*x > 0)){ # Checking whether u_i > 0
    # \forall i = 1, ..., n
    return(length(y)*log(b^a/gamma(a)) +
      (a-1)*sum(log(y - theta[1] - theta[2]*x)) -
      b*sum(y - theta[1] - theta[2] * x))
  }else{
    return(-Inf)
  }
}
```

```
# Initial condition previously defined #
```

```
# The objective function evaluated at the initial condition returns an error.  
# So I'll be subtracting 1
```

```
mod2_true <- optim(par = init_gamma[-3]-1,  
  fn = function(theta){-l_gamma(theta = theta, a = a,  
                                b = b, y = y_2, x = x)})
```

```
print(mod2_true$par)  
## (Intercept)          x  
##    1.495811    2.999999
```

```
# Model 3
```

```
# Objective function
```

```
l_t <- function(theta, y, x){  
  length(y)*(log(gamma((theta[3]+1)/2)) - log(sqrt(theta[3]*pi)) -  
    log(gamma(theta[3]/2))) -  
    (theta[3]+1)/2*sum(log(1 + ((y - theta[1] - theta[2]*x)^2)/theta[3]))  
}
```

```
init_t2 = c(coef(  
  lm(y_3 ~ x)  
) , df = 0.0001) # Setting a pretty small degree of freedom as an initial guess
```

```
mod3_true <- optim(par = init_t2,  
  fn = function(theta){-l_t(theta = theta, y = y_3, x=x)})
```

```
print(mod3_true$par)  
## (Intercept)          x          df  
##    1.4711328    3.0079292    0.9809329
```

```
plot_curves <- function(...){  
  ggplot() +  
    geom_point(aes(y = eval(parse(text = paste0('y_', ...))), x=x), alpha = .5,  
              color = purple_rain_colors[1], shape = 21) +  
    geom_abline(slope = beta[2],  
               intercept = beta[1],  
               linetype = 'F1',  
               color = 'grey',  
               linewidth = 1) +  
    geom_abline(slope = eval(parse(text = paste0('mod', ..., '_true$par[2]'))),  
               intercept = eval(parse(text = paste0('mod', ...,  
                                                     '_true$par[1]'))),  
               linetype = 'dashed',  
               color = purple_rain_colors[1],  
               linewidth = 1,  
               alpha = .75) +  
    geom_abline(slope = eval(parse(text = paste0('mod', ..., '_norm$par[2]'))),
```

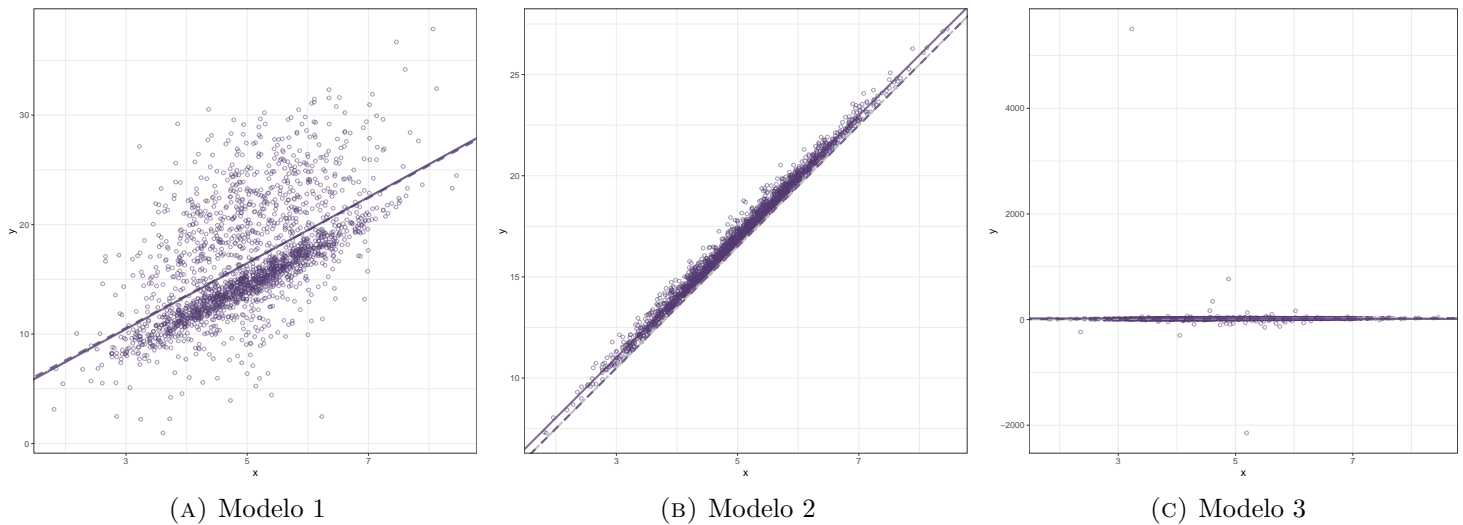


FIGURA 5. Comparativo

```

intercept = eval(parse(text = paste0('mod', ...,
                                      '_norm$par[1]'))),
linetype = 'solid',
color = purple_rain_colors[1],
linewidth = 1,
alpha = .75) +
labs(x = "x", y = "y")
}

invoke_map(.f = plot_curves, .x = 1:3)

```

Na Figura ??, as retas de regressão tracejadas representam as estimativas do Exercício 4, enquanto as retas sólidas representam as estimativas obtidas no Exercício 2. Cabe resultar que a escala dos dados para o modelo 3 omite a incoerência das estimativas supondo-se distribuição gaussiana em relação aos parâmetros verdadeiros,

REFERÊNCIAS

- [1] WH Greene. Econometric analysis, 7th. saddle river, 2011.
- [2] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.
- [4] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Golemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kokske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.
- [5] Yihui Xie. knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch, and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. ISBN 978-1466561595.