

## Tarefa 10

6. Ajuste os dados:

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(ggplot2))

dados <- data.frame(x = c(-8, -6, -4, -2, 0, 2, 4),
                     y = c(30, 10, 9, 6, 5, 4, 4))
print(dados)
```

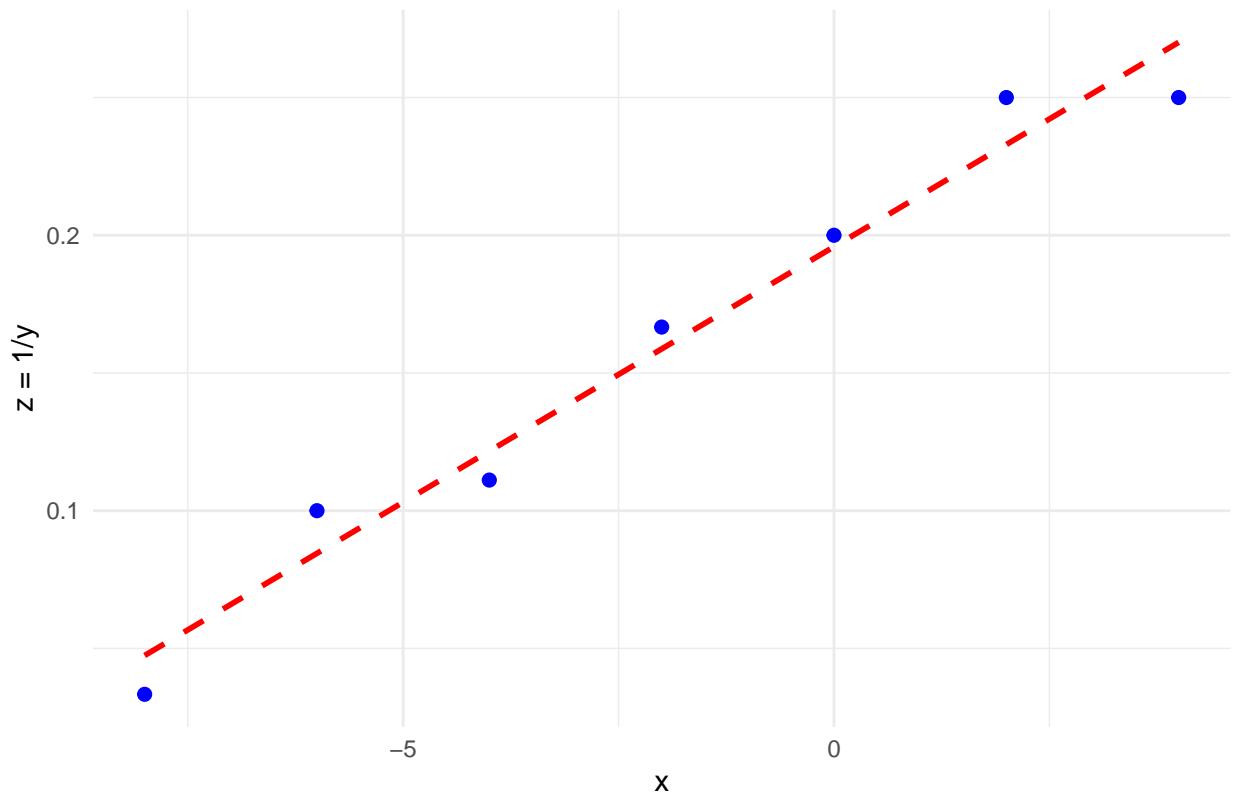
```
##      x   y
## 1 -8 30
## 2 -6 10
## 3 -4  9
## 4 -2  6
## 5  0  5
## 6  2  4
## 7  4  4
```

a) usando a aproximação  $y \approx \frac{y-1}{a_0 + a_1x}$ . Faça o gráfico para  $1/y$  e verifique que esta aproximação é viável;

```
dados_a <- dados %>% mutate(z = 1 / y)

plot_linear_a <- ggplot(dados_a, aes(x = x, y = z)) +
  geom_point(color = "blue", size = 2) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Verificação de Linearidade para o Ajuste (a)", x = "x", y = "z = 1/y") +
  theme_minimal()
print(plot_linear_a)
```

## Verificação de Linearidade para o Ajuste (a)



```
modelo_a <- lm(z ~ x, data = dados_a)
coef_a <- coef(modelo_a)
a0 <- coef_a[1]
a1 <- coef_a[2]

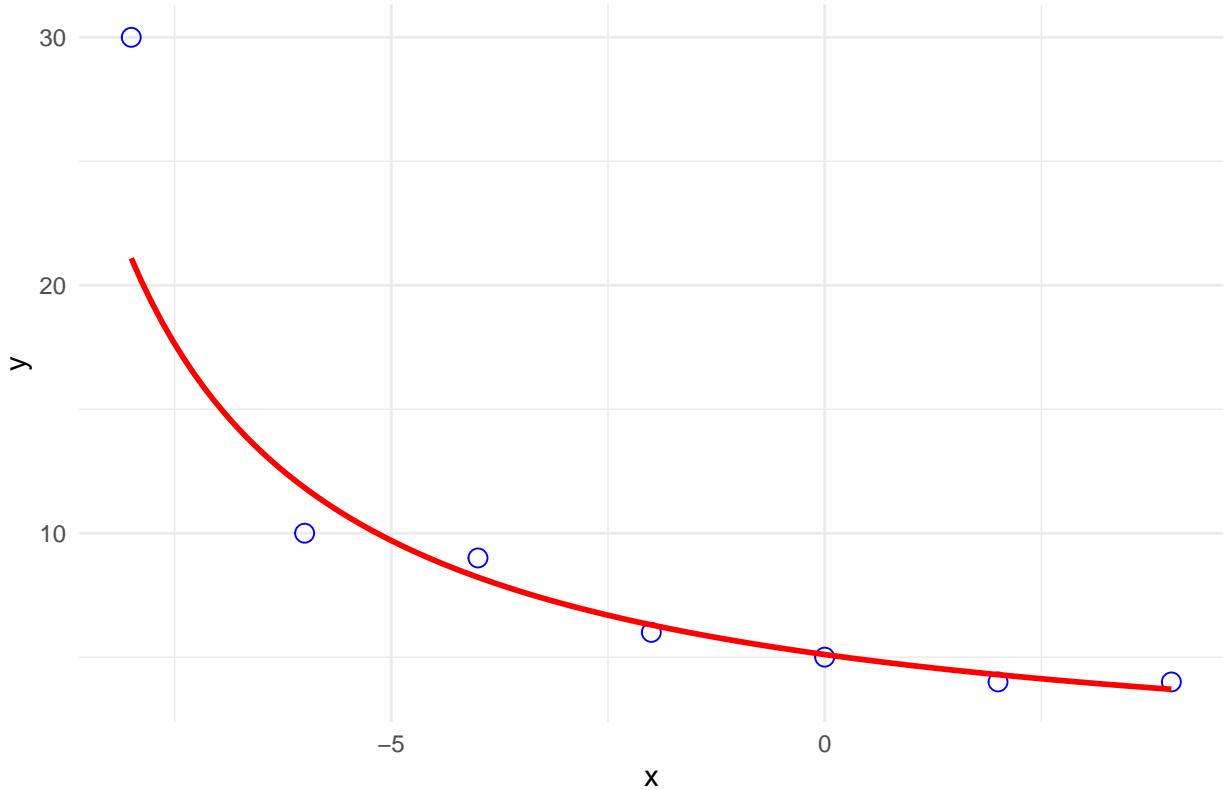
cat(sprintf("Modelo linearizado: z = %.4f + %.4f * x", a0, a1))
```

```
## Modelo linearizado: z = 0.1958 + 0.0186 * x
```

```
phi_a <- function(x_val) { 1 / (a0 + a1 * x_val) }

plot_ajuste_a <- ggplot(dados, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 3, shape=1) +
  stat_function(fun = phi_a, color = "red", linewidth = 1) +
  labs(title = "Ajuste (a): y = 1 / (a0 + a1*x)", x = "x", y = "y") +
  theme_minimal()
print(plot_ajuste_a)
```

Ajuste (a):  $y = 1 / (a_0 + a_1 * x)$

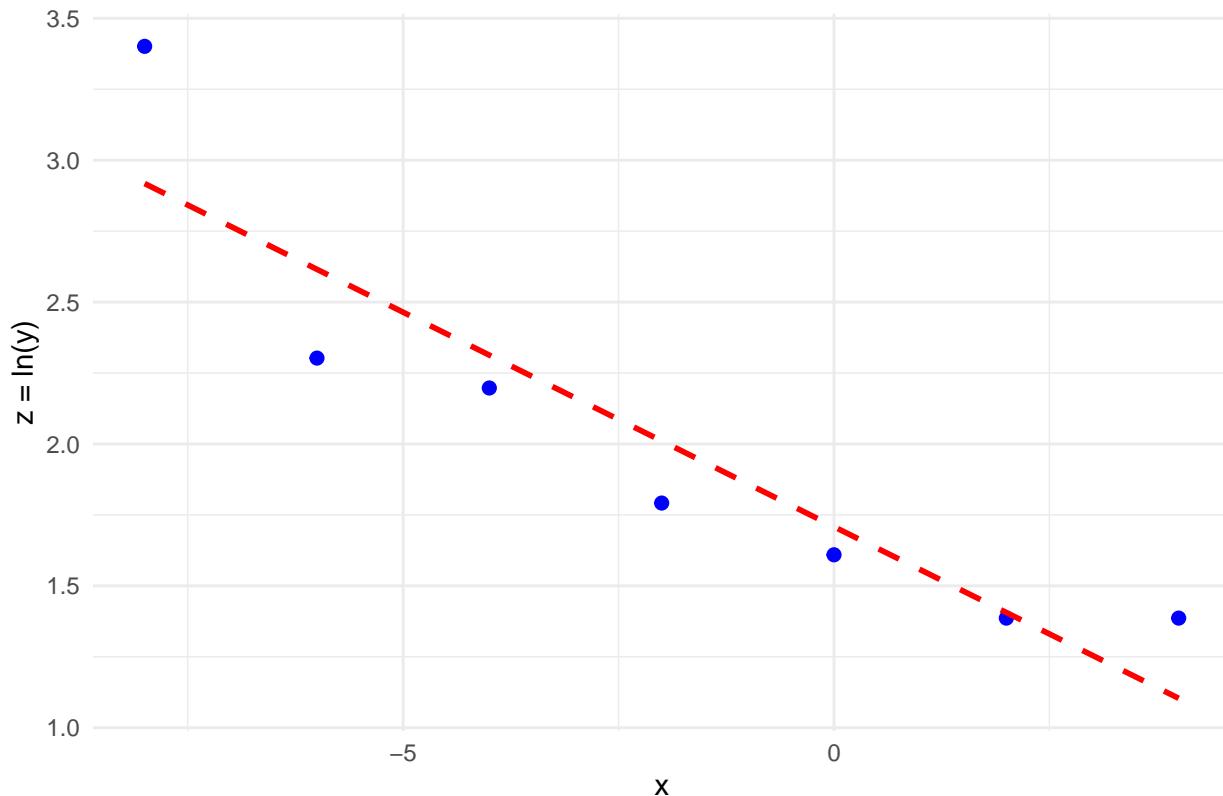


b) idem para  $y \approx ab^x$ ;

```
dados_b <- dados %>% mutate(z = log(y))

plot_linear_b <- ggplot(dados_b, aes(x = x, y = z)) +
  geom_point(color = "blue", size = 2) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Verificação de Linearidade para o Ajuste (b)", x = "x", y = "z = ln(y)") +
  theme_minimal()
print(plot_linear_b)
```

## Verificação de Linearidade para o Ajuste (b)



```

modelo_b <- lm(z ~ x, data = dados_b)
coef_b <- coef(modelo_b)
a0_ln <- coef_b[1]
a1_ln <- coef_b[2]

a <- exp(a0_ln)
b_exp <- exp(a1_ln)

cat(sprintf("Modelo linearizado: z = %.4f + %.4f * x", a0_ln, a1_ln))

```

```
## Modelo linearizado: z = 1.7084 + -0.1512 * x
```

```
cat(sprintf("Coeficientes originais estimados: a = %.4f, b = %.4f", a, b_exp))
```

```
## Coeficientes originais estimados: a = 5.5199, b = 0.8597
```

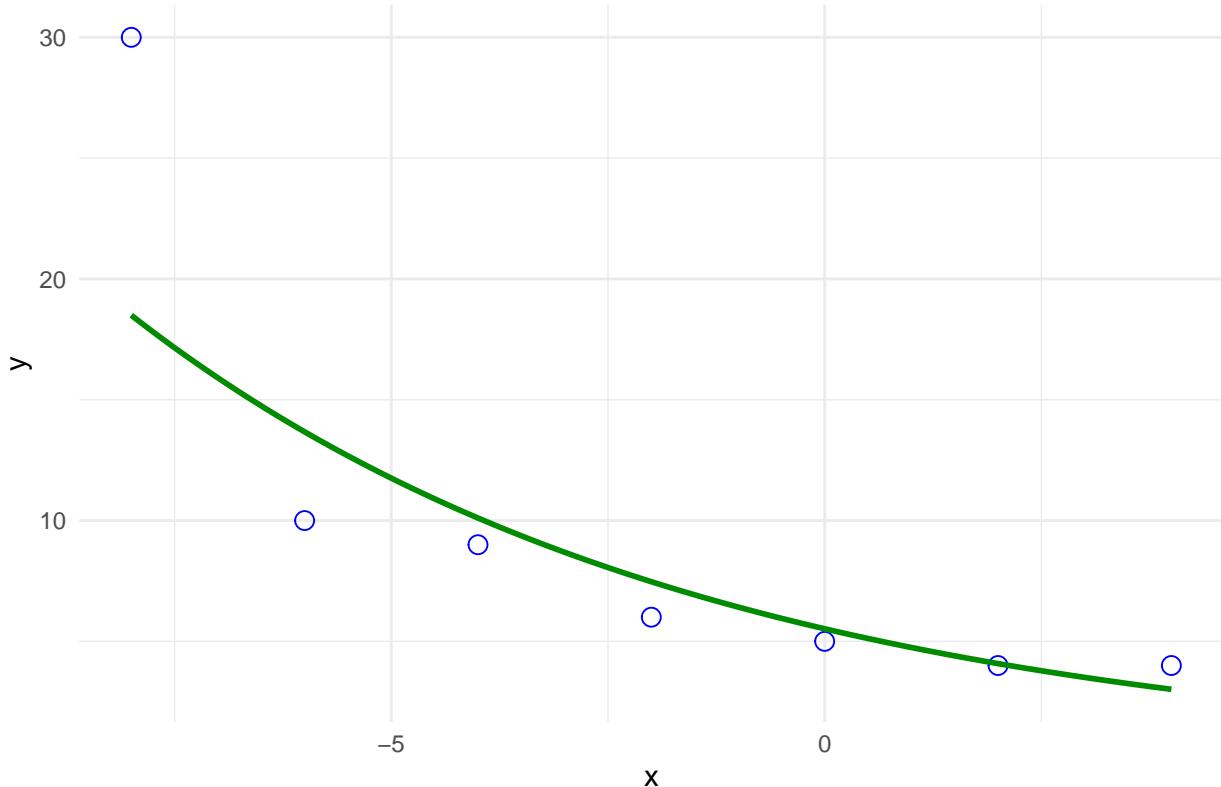
```

phi_b <- function(x_val) { a * (b_exp^x_val) }

plot_ajuste_b <- ggplot(dados, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 3, shape = 1) +
  stat_function(fun = phi_b, color = "green4", linewidth = 1) +
  labs(title = "Ajuste (b): y = a * b^x", x = "x", y = "y") +
  theme_minimal()
print(plot_ajuste_b)

```

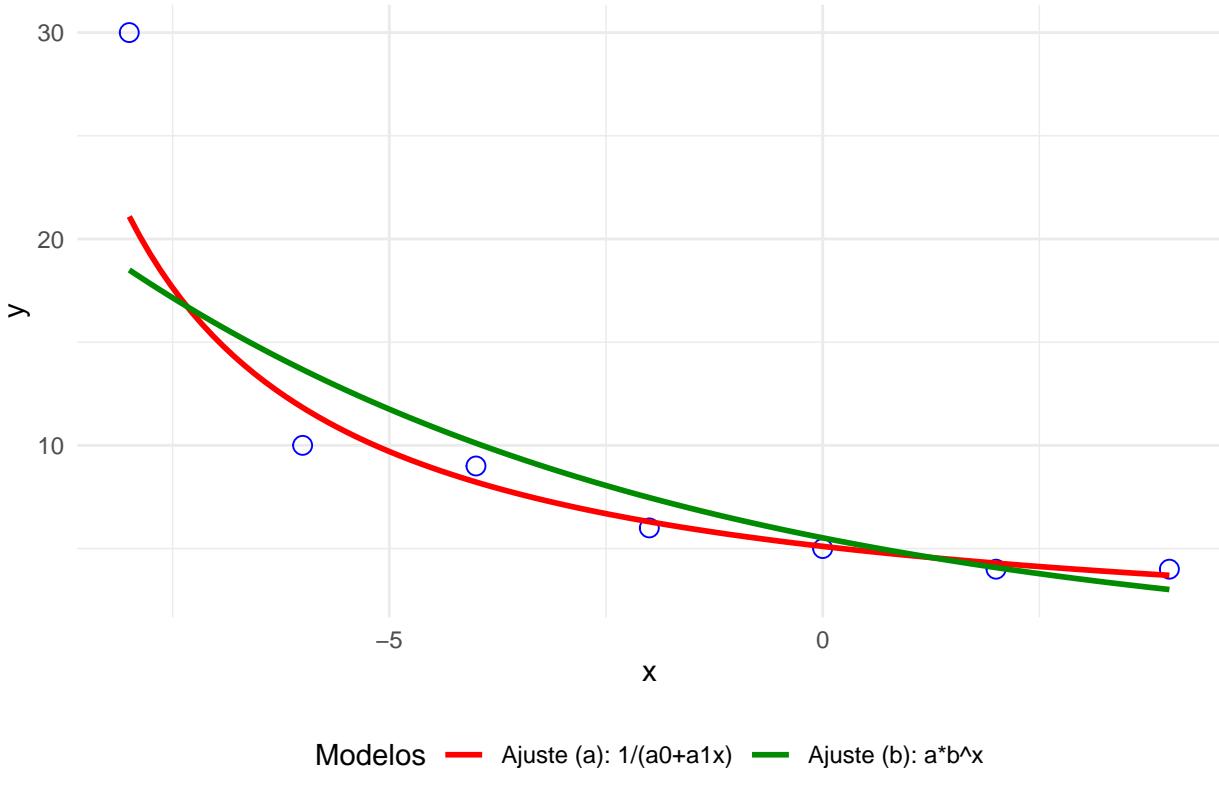
Ajuste (b):  $y = a \cdot b^x$



c) compare os resultados (a) e (b).

```
plot_comparativo <- ggplot(dados, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 3, aes(label = "Dados Originais")) +
  stat_function(fun = phi_a, aes(color = "Ajuste (a): 1/(a0+a1x)"), linewidth = 1) +
  stat_function(fun = phi_b, aes(color = "Ajuste (b): a*b^x"), linewidth = 1) +
  scale_color_manual(name = "Modelos", values = c("red", "green4")) +
  labs(title = "Comparação dos Ajustes", x = "x", y = "y") +
  theme_minimal() + theme(legend.position = "bottom")
print(plot_comparativo)
```

## Comparação dos Ajustes



```
sse_a <- sum((dados$y - phi_a(dados$x))^2)
sse_b <- sum((dados$y - phi_b(dados$x))^2)

cat(sprintf("SSE para o Ajuste (a) [y = 1/(a0+a1x)]: %.4f", sse_a))
```

```
## SSE para o Ajuste (a) [y = 1/(a0+a1x)]: 83.6594
```

```
cat(sprintf("SSE para o Ajuste (b) [y = a*b^x] : %.4f", sse_b))
```

```
## SSE para o Ajuste (b) [y = a*b^x] : 150.4032
```

O ajuste (a) apresentou menor SSE na escala original e portanto deve ser o escolhido para essa aproximação.

7. O número de bactérias, por unidade de volume, existente em uma cultura após x horas é apresentado na tabela:

```
dados_bac <- data.frame(x = 0:6,
                           y = c(32, 47, 65, 92, 132, 190, 275))
print(dados_bac)
```

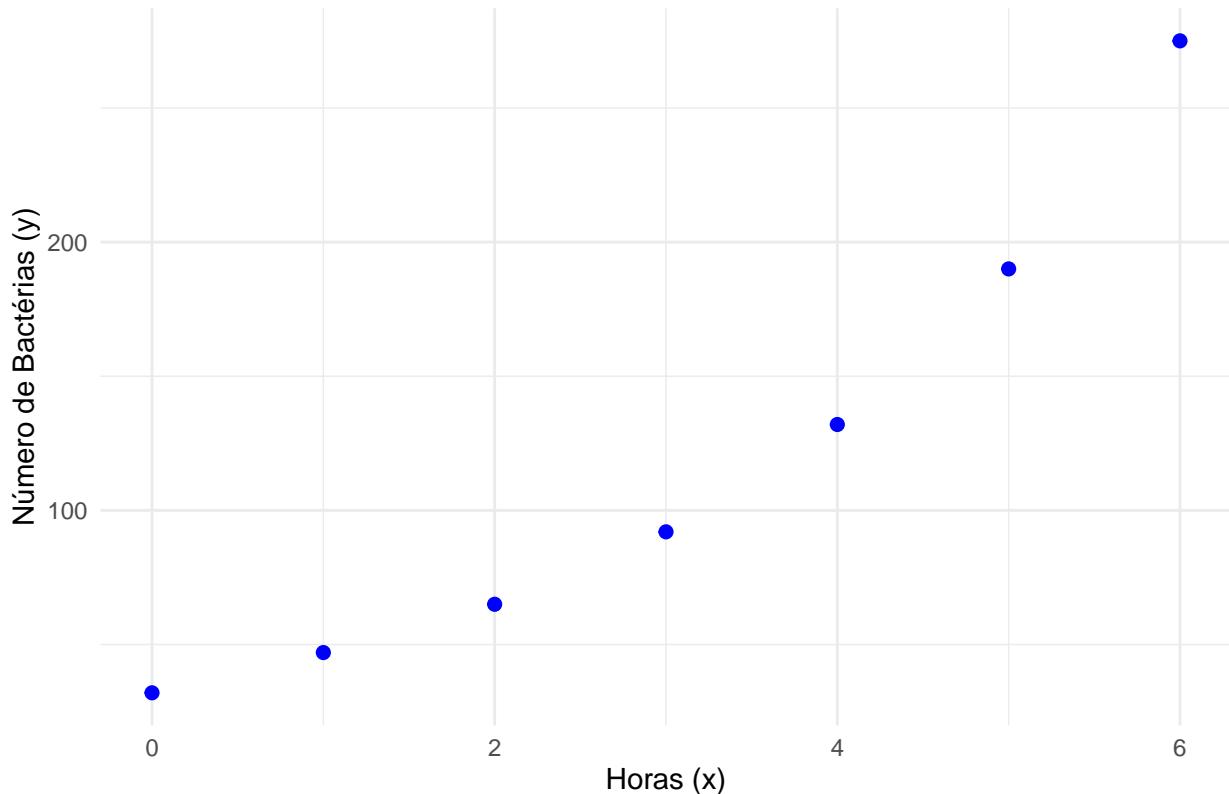
```
##   x   y
## 1 0  32
## 2 1  47
```

```
## 3 2 65
## 4 3 92
## 5 4 132
## 6 5 190
## 7 6 275
```

a) verifique que uma curva para se ajustar ao diagrama de dispersão é do tipo exponencial;

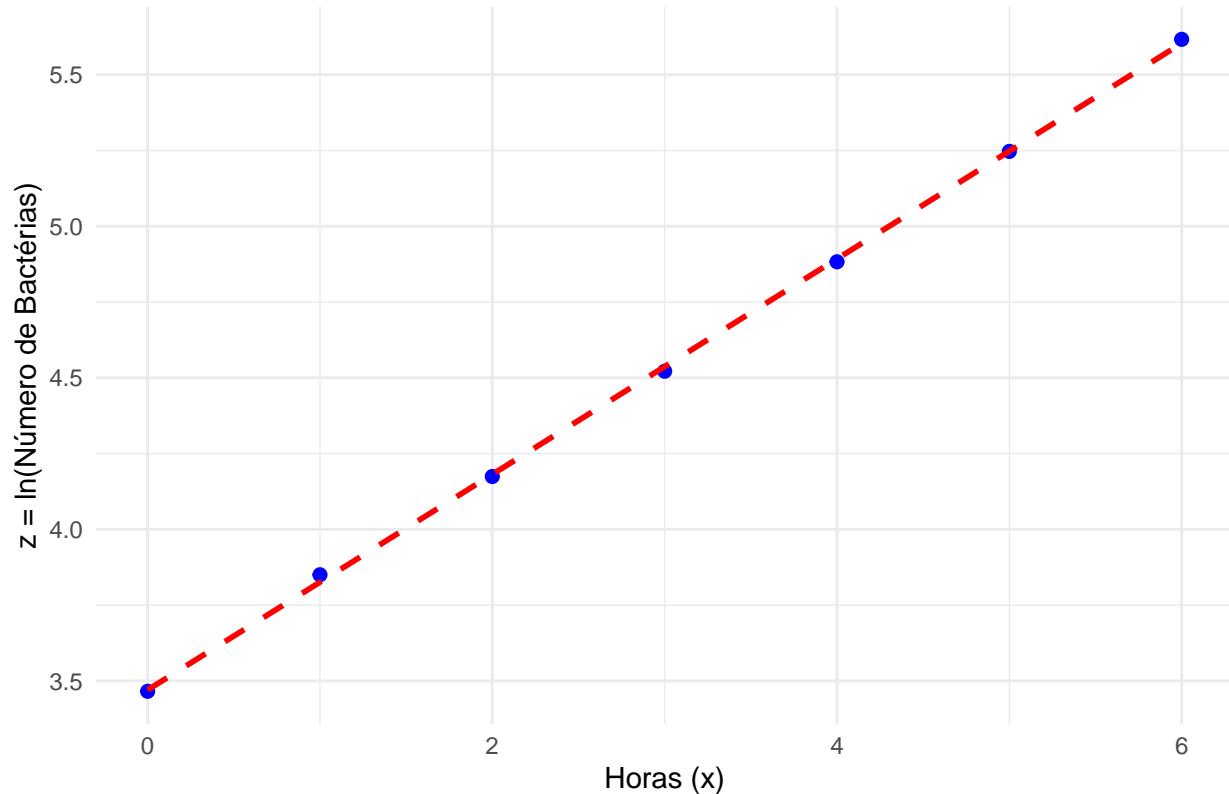
```
plot_dispersao <- ggplot(dados_bac, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 2) +
  labs(title = "Diagrama de Dispersão Original",
       x = "Horas (x)", y = "Número de Bactérias (y)") +
  theme_minimal()
print(plot_dispersao)
```

Diagrama de Dispersão Original



```
dados_exp_linear <- dados_bac %>% mutate(z = log(y))
plot_semilog <- ggplot(dados_exp_linear, aes(x = x, y = z)) +
  geom_point(color = "blue", size = 2) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Verificação de Linearidade (Semi-log)",
       x = "Horas (x)", y = "z = ln(Número de Bactérias)") +
  theme_minimal()
print(plot_semilog)
```

## Verificação de Linearidade (Semi-log)



- b) ajuste aos da.dos as curvas  $y = ab^x$  e  $y = ax^b$ ; compare os valores obtidos por meio destas equações com os dados experimentais;

```

modelo_exp <- lm(z ~ x, data = dados_exp_linear)
coef_exp <- coef(modelo_exp)
a0_ln <- coef_exp[1]
a1_ln <- coef_exp[2]

a_exp <- exp(a0_ln)
b_exp <- exp(a1_ln)

cat(sprintf("Modelo linearizado: ln(y) = %.4f + %.4f * x", a0_ln, a1_ln))

## Modelo linearizado: ln(y) = 3.4703 + 0.3555 * x

cat(sprintf("Coeficientes originais estimados: a = %.4f, b = %.4f", a_exp, b_exp))

## Coeficientes originais estimados: a = 32.1469, b = 1.4270

phi_exp <- function(x_val) { a_exp * (b_exp^x_val) }
sse_exp <- sum((dados_bac$y - phi_exp(dados_bac$x))^2)
cat(sprintf("SSE (escala original): %.4f", sse_exp))

## SSE (escala original): 18.1421

```

```

dados_pot_linear <- dados_bac %>%
  filter(x > 0) %>%
  mutate(t = log(x), w = log(y))

modelo_pot <- lm(w ~ t, data = dados_pot_linear)
coef_pot <- coef(modelo_pot)
c0_ln <- coef_pot[1]
c1 <- coef_pot[2]

a_pot <- exp(c0_ln)
b_pot <- c1

cat(sprintf("Modelo linearizado: ln(y) = %.4f + %.4f * ln(x)", c0_ln, c1))

## Modelo linearizado: ln(y) = 3.6594 + 0.9631 * ln(x)

cat(sprintf("Coeficientes originais estimados: a = %.4f, b = %.4f", a_pot, b_pot))

## Coeficientes originais estimados: a = 38.8387, b = 0.9631

phi_pot <- function(x_val) { ifelse(x_val > 0, a_pot * (x_val^b_pot), NA) }

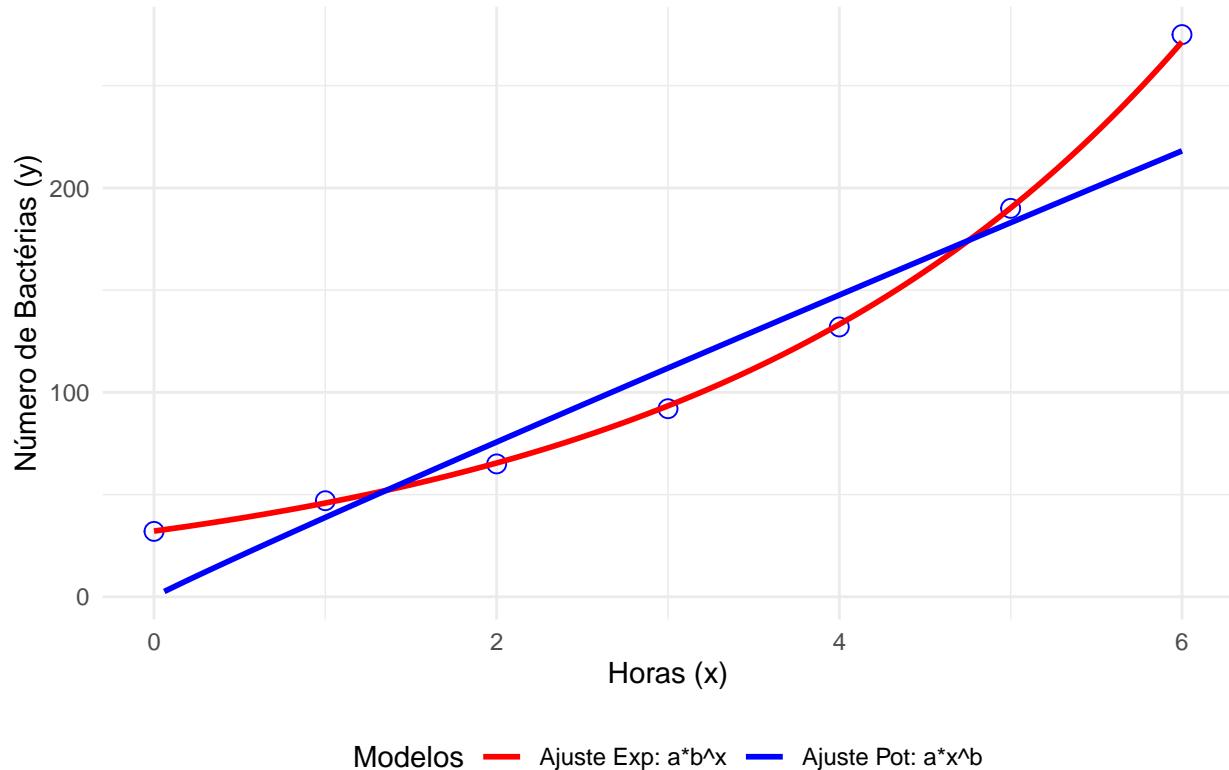
sse_pot <- sum((dados_bac$y[dados_bac$x > 0] -
                 phi_pot(dados_bac$x[dados_bac$x > 0]))^2)
cat(sprintf("SSE (escala original, x > 0): %.4f", sse_pot))

## SSE (escala original, x > 0): 4104.8645

plot_comparativo_bacterias <- ggplot(dados_bac, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 3, shape = 1) +
  stat_function(fun = phi_exp, aes(color = "Ajuste Exp: a*b^x"), linewidth = 1) +
  stat_function(fun = phi_pot, aes(color = "Ajuste Pot: a*x^b"), linewidth = 1, na.rm = TRUE) +
  scale_color_manual(name = "Modelos", values = c("red", "blue")) +
  labs(title = "Comparação dos Ajustes para Crescimento Bacteriano",
       x = "Horas (x)", y = "Número de Bactérias (y)") +
  theme_minimal() + theme(legend.position = "bottom")
print(plot_comparativo_bacterias)

```

## Comparação dos Ajustes para Crescimento Bacteriano



```
cat(sprintf("Comparação SSE (x > 0): Exp = %.4f, Pot = %.4f", sse_exp, sse_pot))
```

```
## Comparação SSE (x > 0): Exp = 18.1421, Pot = 4104.8645
```

Conclusão: O ajuste exponencial  $y = ab^x$  apresentou menor SSE na escala original (para  $x>0$ ).

c) avalie da melhor forma o valor de  $y(x)$  para  $x = 7$ .

```
y_estimado_7 <- phi_exp(7)
cat(sprintf("Usando o modelo y = %.4f * (%.4f)^x:", a_exp, b_exp))
```

```
## Usando o modelo y = 32.1469 * (1.4270)^x:
```

```
cat(sprintf("O número estimado de bactérias após 7 horas é: %.0f\n", y_estimado_7))
```

```
## O número estimado de bactérias após 7 horas é: 387
```