

---

**Exercício 2****Guilherme Vinícius Amorim**

---

**1 Problema Não-Linearmente Separável**

Um problema linearmente separável é um programa cuja solução pode ser obtida dividindo-se o espaço de entrada em duas regiões através de uma reta. O problema a seguir não é linearmente separável:

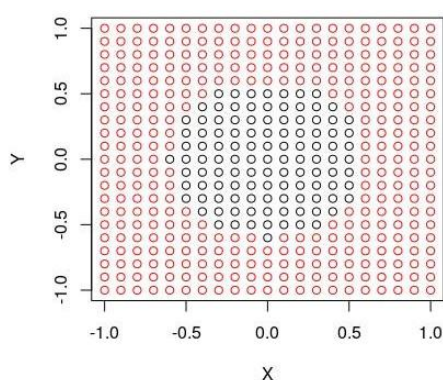


Figura 1: Pontos em vermelho pertencem à classe positiva e pontos em preto pertencem à classe negativa

Contudo, a partir de uma projeção não linear, o sistema pode se transformar em linearmente separável. O mapa de calor abaixo demonstra como podemos separar os pontos em relação à proximidade em relação aos eixos X e Y:

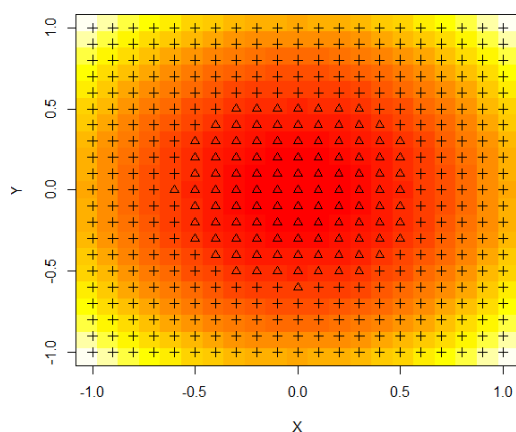


Figura 2: Mapa de calor

A partir da seguinte rotina em R, uma mudança de estado possibilitou tornar o problema linearmente separável:

```
#Projeção não linear:
mudancaDeEspaco = function(x,y){
  return(x^2 + y^2)
}

Resp = mudancaDeEspaco(create_grid$var1,create_grid$var2)
plot(Resp, col = classe+1)
abline(h = raio^2)
```

Figura 3: Mudança de espaço em R tornando o problema linearmente separável

A seguinte imagem demonstra como essa projeção não linear tornou o problema linearmente separável:

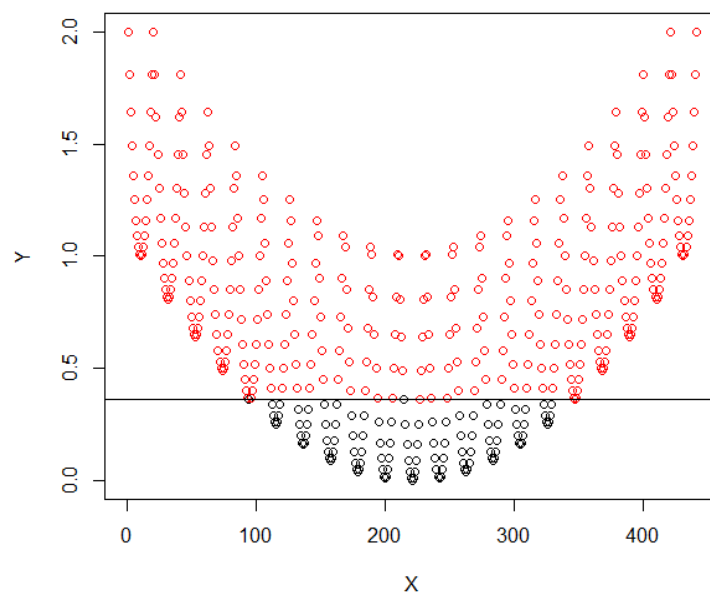


Figura 4: Nova projeção do sistema que agora é linearmente separável

## 2 Overfitting e Underfitting

- Qual dos 3 modelos construídos (preto, vermelho e azul) parece ser o mais adequado aos dados, sabendo-se que existe um ruído na amostragem?

O modelo mais adequado aos dados é o modelo azul, uma vez que considera-se que durante a amostragem existe ruído.

- Qual dos modelos apresenta menor erro de treinamento?

O modelo com menor erro de treinamento é o modelo preto, uma vez que, neste modelo, a função gerada contém a maioria dos pontos utilizados no treinamento, possuindo um erro em relação às amostras quase nulo.

- Quais os problemas com cada um dos modelos inadequados?

No modelo em preto, o erro em relação às amostras é praticamente nulo. Contudo, não levou-se em consideração o ruído, fazendo com que o modelo não seja o mais adequado para representar um problema de regressão. No modelo em vermelho, o grau do polinômio gerado é grau 1, não se aproximando do que se foi amostrado e tendo um erro em alguns pontos muito elevado. Dessa forma, no modelo em preto ocorre um *Overfit*, dado que este usa um número excessivo de neurônios, enquanto no modelo em vermelho ocorre um *Underfit*, dado que este usa um número não suficiente de neurônios no treinamento.

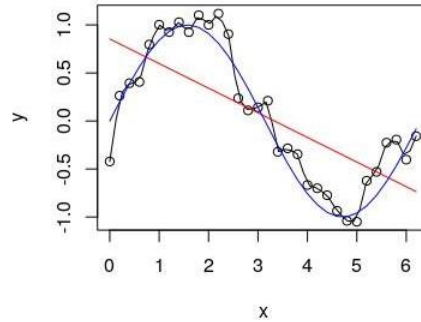


Figura 5: Ajuste de três modelos para um problema de regressão

### 3 Aproximação Polinomial

- Obtenha aproximações polinomiais a partir de 10 amostras da função geradora  $f_g(x) = \frac{1}{2}x^2 + 3x + 10$  somadas com um ruído gaussiano  $N(mean = 0, sd = 4)$  amostradas entre  $x = -15$  e  $x = 10$ , com um número de amostras  $N = 20$  e grau do polinômio variando entre  $p = 1$  a  $p = 8$ . Para cada aproximação, mostre um gráfico com a função geradora, as amostras e o polinômio obtido.

Para obtermos uma aproximação polinomial  $p(x)$  a partir da função geradora  $f(x)$  consideramos:

$$p(x) = w_px^p + w_{p-1}x^{p-1} + \dots + w_1x + w_0 \quad (1)$$

em que  $x$  é o argumento e  $w_i$  é o coeficiente do termo de grau  $i$ .

Os parâmetros de  $p(x)$  são ajustados a partir do seguinte sistema de equações:

$$\begin{array}{ccccccc} y_1 & = & w_px_1^p & + & w_{p-1}x_1^{p-1} & + & \dots & + & w_1x_1 & + & w_0 \\ y_2 & = & w_px_2^p & + & w_{p-1}x_2^{p-1} & + & \dots & + & w_1x_2 & + & w_0 \\ \vdots & & \vdots & & \vdots & & \dots & & \vdots & & \\ y_N & = & w_px_N^p & + & w_{p-1}x_N^{p-1} & + & \dots & + & w_1x_N & + & w_0 \end{array} \quad (2)$$

O sistema acima pode ser representado na forma matricial por:

$$\mathbf{H}\mathbf{w} = \mathbf{y} \quad (3)$$

E  $\mathbf{H}$ ,  $\mathbf{w}$  e  $\mathbf{y}$  são representados por:

$$\mathbf{H} = \begin{bmatrix} x_1^p & x_1^{p-1} & \cdots & x_1 & 1 \\ x_2^p & x_2^{p-1} & \cdots & x_2 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_N^p & x_N^{p-1} & \cdots & x_N & 1 \end{bmatrix} \quad (4)$$

$$\mathbf{w} = \begin{bmatrix} w_p \\ w_{p-1} \\ \vdots \\ w_1 \\ w_0 \end{bmatrix} \quad (5)$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (6)$$

Para obter as 10 amostras da função geradora, a seguinte rotina foi implementada em R:

```
geradora <- function(x){
  return(0.5*x^2+3*x+10)
}
# Gerando 10 amostras da função geradora somadas com um ruído gaussiano
x <- runif(N_amostras, min = -15, max = 10)
y = geradora(x) + rnorm(n = N_amostras, sd = 4)
plot(x, y, col = "red") # Plot amostras
```

Figura 6: Gerando 10 amostras da função geradora

A partir das 10 amostras geradas, o primeiro passo para solucionar o problema dado foi gerar a matriz H, que é representada por (4). Isso foi implementado da seguinte forma:

```
poly_fun <- function(x, grau){
  x_vec <- c(1)
  for(p in 1:grau){
    x_vec <- c(x^p, x_vec)
  }
  return(x_vec)
}
# Gerando H:
H <- matrix(0, nrow = N_amostras, ncol = grau + 1)
for(i in 1 : N_amostras){
  H[i,] <- poly_fun(x[i],grau)
}
```

Figura 7: Gerando a matriz H

O próximo passo foi gerar a matriz w, que é representada por (5). Como a matriz H já foi gerada, assim como y já é conhecido a partir do momento que as amostras foram geradas, temos que:

$$\mathbf{w} = \mathbf{H}^+ \mathbf{y} \quad (7)$$

em que  $\mathbf{H}^+$  é a pseudo inversa de H. Portanto, a seguinte rotina em R realizou essa operação:

```
# Calculando w:
w <- pseudoinverse(H)%*%y # w = H+ y
```

Figura 8: Calculando a matriz w

Como já se é conhecido as matrizes H, w e y do sistema matricial (3), o último passo foi gerar as aproximações polinomiais. Isso foi feito a partir da seguinte rotina:

```
poly_fun <- function(x, grau){
  x_vec <- c(1)
  for(p in 1:grau){
    x_vec <- c(x^p, x_vec)
  }
  return(x_vec)
}

hat_calc <- function(x, w){
  poly <- poly_fun(x, length(w) - 1)
  return(poly%*%w)
}

# Gerando o polinomio aproximado:
seq_x <- seq(-15, 10, 0.2)
y_hat <- c()
for(i in 1 : length(seq_x)){
  y_hat[i] <- hat_calc(seq_x[i], w)
}
```

Figura 9: Gerando as aproximações polinomiais

Os gráficos abaixo comparam a função geradora, as amostras e o polinômio aproximado:

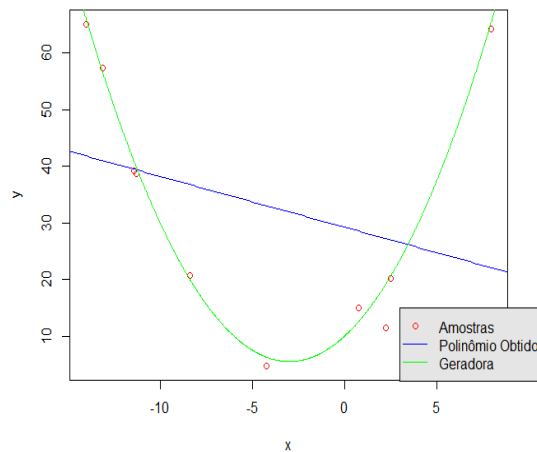


Figura 10: Aproximação polinomial de grau 1

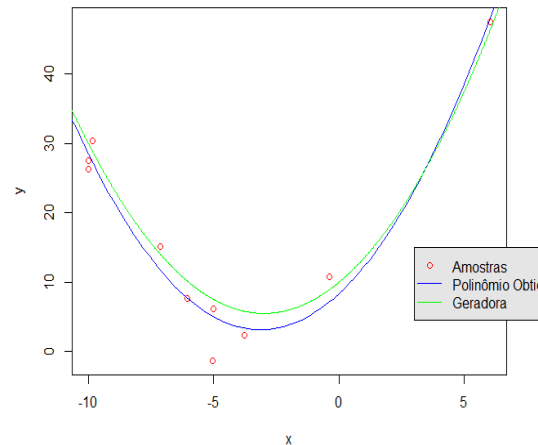


Figura 11: Aproximação polinomial de grau 2

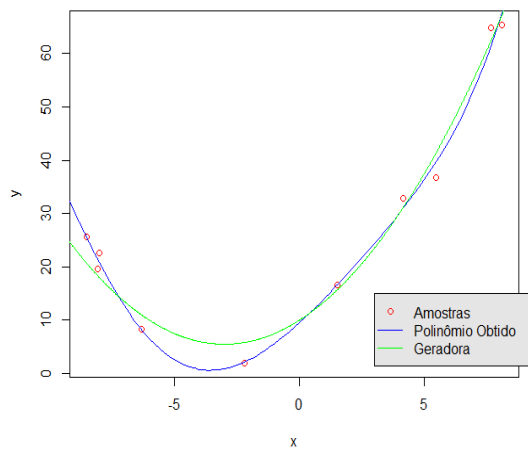


Figura 12: Aproximação polinomial de grau 3

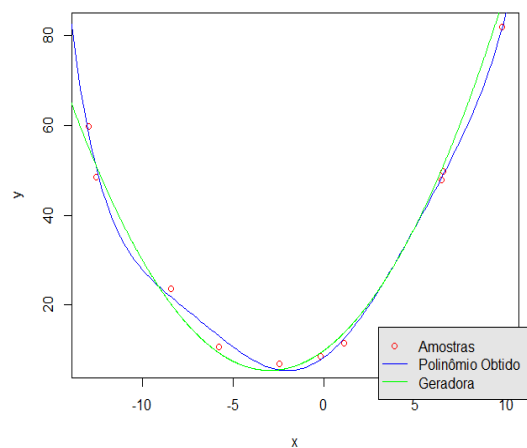


Figura 13: Aproximação polinomial de grau 4

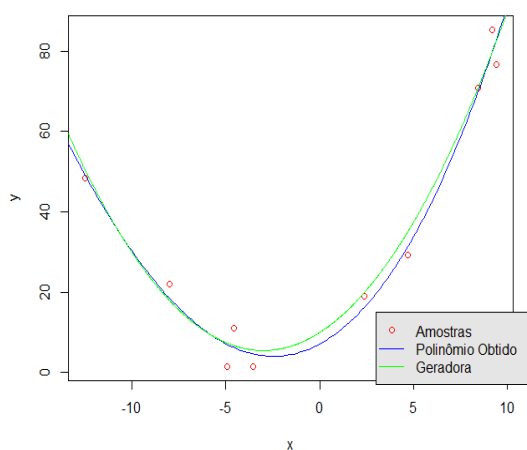


Figura 14: Aproximação polinomial de grau 5

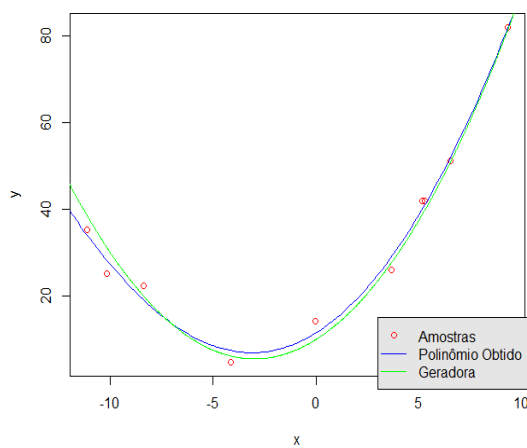


Figura 15: Aproximação polinomial de grau 6

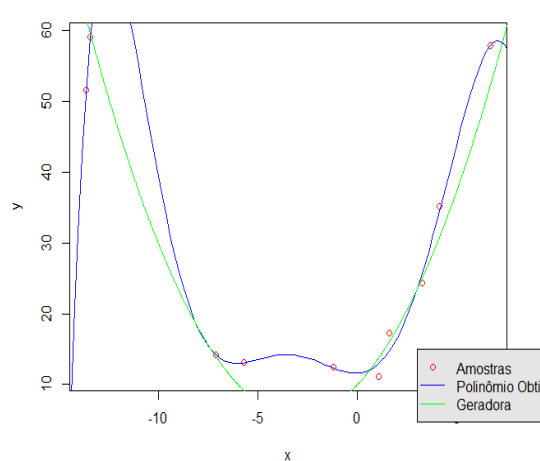


Figura 16: Aproximação polinomial de grau 7

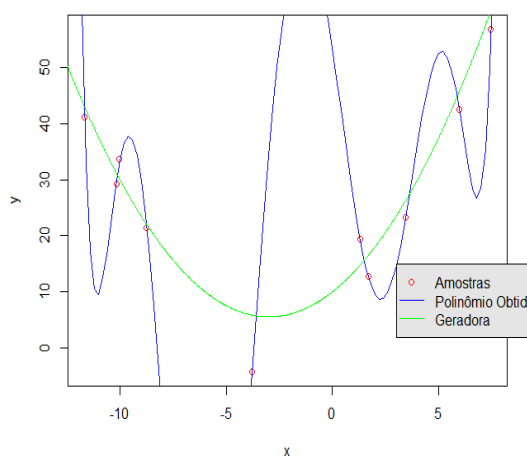


Figura 17: Aproximação polinomial de grau 8

- Responda: Ocorreu *overfitting*? Ocorreu *underfitting*? Em quais casos ocorreu estes fenômenos?

O modelo *overfitting* ocorre quando o modelo se adaptou muito bem aos dados com os quais está sendo treinado; porém, não generaliza bem para novos dados. Ou seja, o modelo “decorou” o conjunto de dados de treino, mas não aprendeu de fato o que diferencia aqueles dados para quando precisar enfrentar novos testes. Na minha opinião, ocorreu *overfitting* com os polinômios aproximados de grau 7 e 8.

O modelo *underfitting* ocorre quando o modelo não se adapta bem sequer aos dados com os quais foi treinado. Ocorreu *underfitting* com o polinômio aproximado de grau 1 em função do grau utilizado na aproximação ter sido abaixo do necessário.

- Repita o procedimento para 100 amostras ao invés de 10. Qual o impacto do número de amostras na aproximação polinomial?

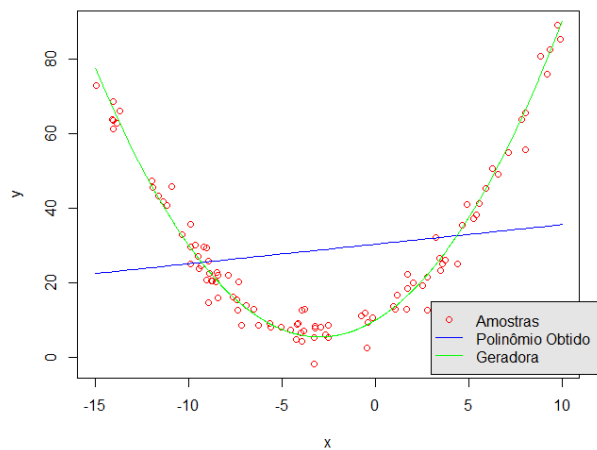


Figura 18: Aproximação polinomial de grau 1

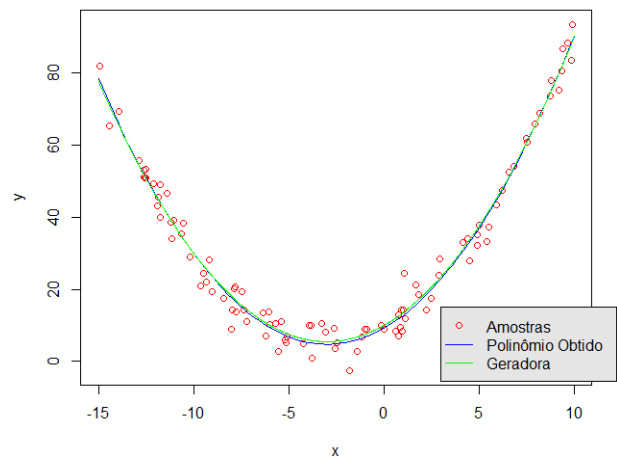


Figura 19: Aproximação polinomial de grau 2

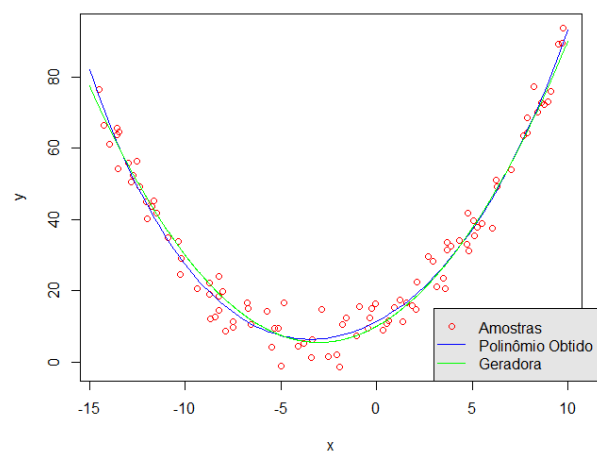


Figura 19: Aproximação polinomial de grau 3

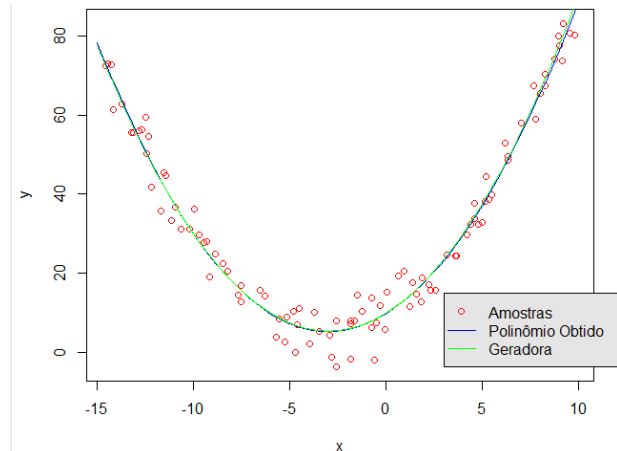


Figura 20: Aproximação polinomial de grau 4

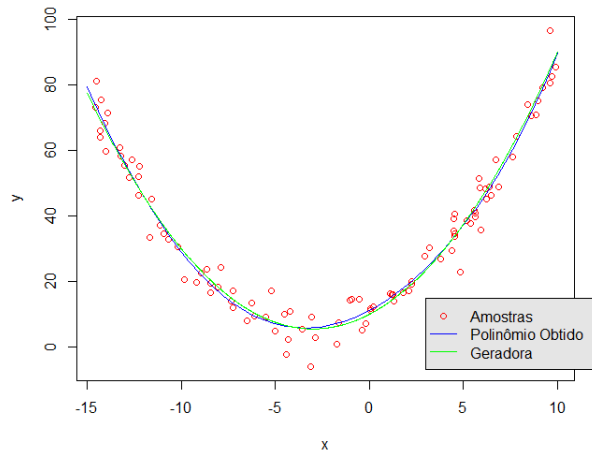


Figura 21: Aproximação polinomial de grau 5

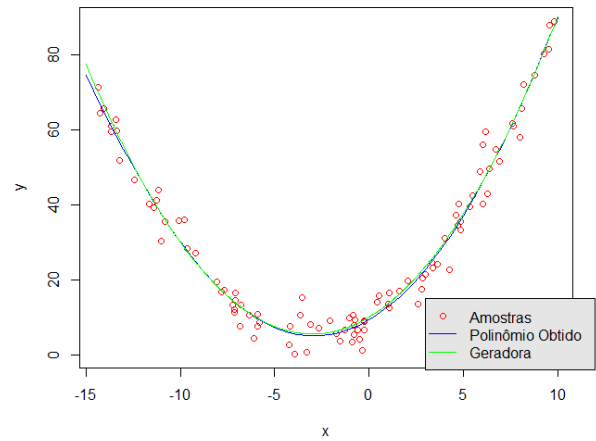


Figura 22: Aproximação polinomial de grau 6

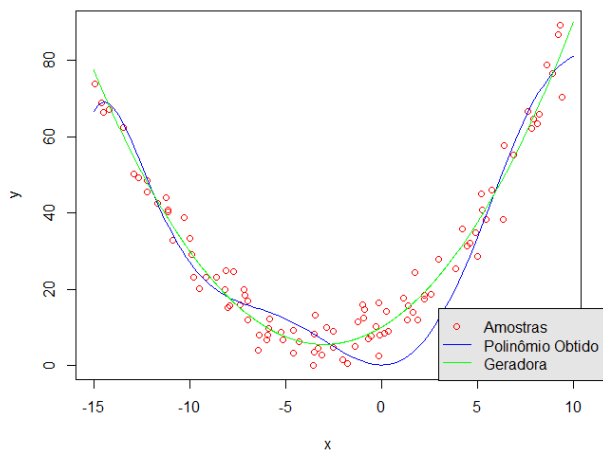


Figura 23: Aproximação polinomial de grau 7

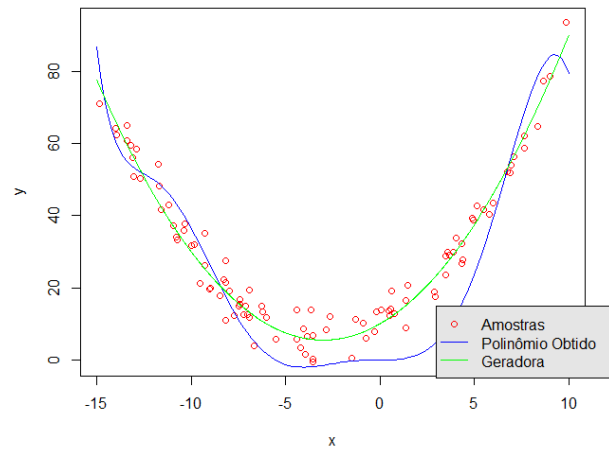


Figura 24: Aproximação polinomial de grau 8

Observa-se que o aumento de amostras da função geradora acarreta em uma melhor aproximação polinomial. Com 10 amostras, nos polinômios de grau 7 e 8 ocorreu *overfitting*, enquanto no polinômio de grau 1 ocorreu *underfitting*. O mesmo fenômeno ocorreu com 100 amostras. Todavia, para os polinômios em que se ocorreu *overfitting*, a aproximação polinomial se aproximou mais da função geradora. Isso mostra que mesmo que haja uma escolha equivocada no grau do polinômio da função aproximada, um alto número de amostras pode compensar essa má escolha e fazer com que a função aproximada se aproxime da função geradora.