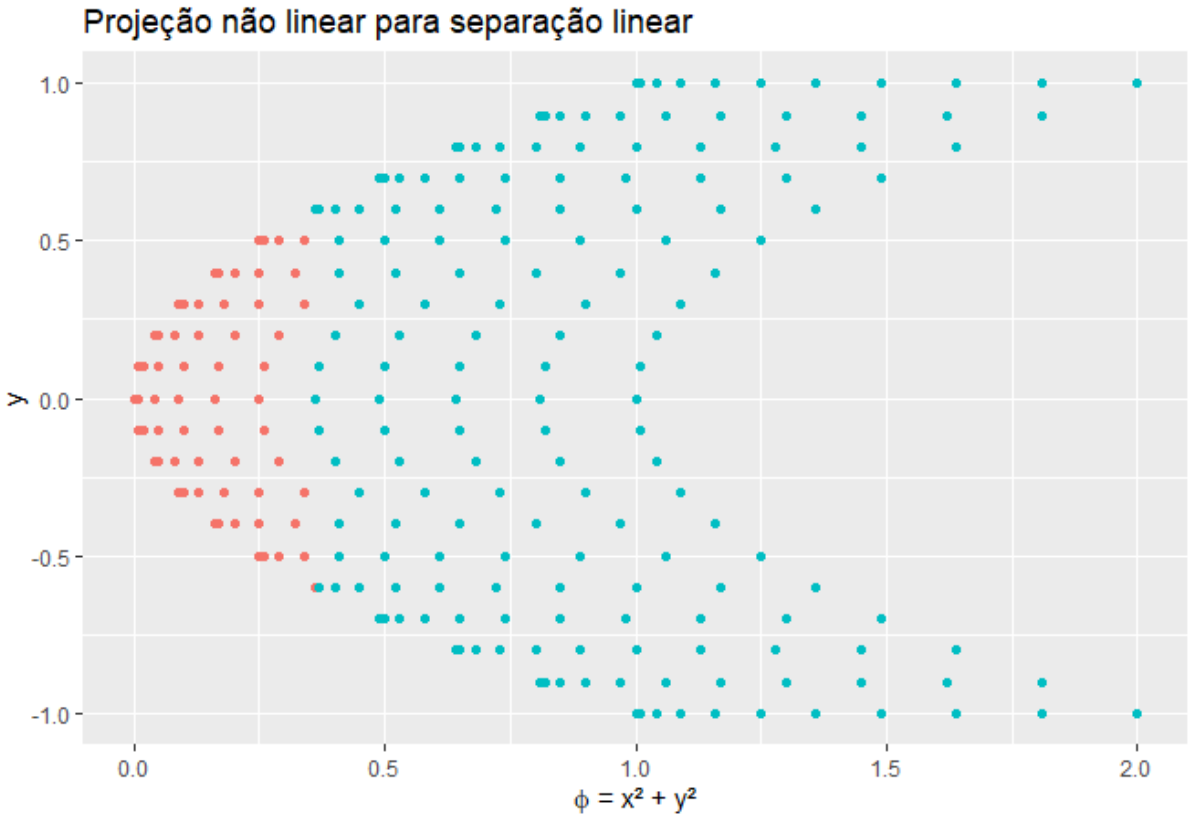
Exercício 2.
Nome: Lucas Kou Kinoshita
RM: 2019021557
data: 22/03/2025

## 1. Problema não-linearmente separável

Os pontos foram mapeados para num novo espaço $\phi(x, y) = x^2 + y^2$, gerando uma projeção não-linear que torne o problema linearmente separável. Como podemos observar no trecho de código e gráfico abaixo:

```r
# Definição das classes
classe_labels <- 1 * (circle(grid$x, grid$y) > raio)

grid$classe <- as.factor(classe_labels)  # Evita o erro

# Projeção não linear (transformacao quadrática)
grid$phi <- grid$x^2 + grid$y^2

# Visualização
library(ggplot2)
ggplot(grid, aes(x = phi, y = y, color = classe)) +
  geom_point() +
  labs(title = "Projeção não linear para separação linear",
       x = expression(phi~"= x² + y²"),
       y = "y")
```



## 2. Overfitting e underfitting

- O modelo azul parece ser a melhor aproximação da função geradora, já que não é exageradamente afetado pelo ruído de amostragem e não parece ser simplificado demais.
- O modelo preto apresenta o menor erro de treinamento, já que é super sensível à variações nos dados de treino.
- O modelo azul deve ter melhor desempenho para novos dados por ser - provavelmente - uma aproximação mais fiel da função geradora.
- Um baixo erro de treinamento não necessariamente implica um melhor desempenho a longo prazo. Já que pode significar uma super sensibilidade do modelo a variações nos dados de treinamento amostrados, o que significa que este não estaria bem comportado para outras variações de dados amostrados pela mesma função geradora, ou seja, pode caracterizar um overfitting.

## 3. Aproximação polinomial

Implementação:
Os dados são gerados de acordo com o enunciado, com função geradora $fg(x) = 1$
$2\,x2 + 3x + 10$, ruído com desvio padrão igual à 4 centralizado em 0, amostrado entre
$x = [-15, 10]$.

```
fgx <- function(xin) 0.5*xin^2 + 3*xin + 10 # Função geradora fg(x)

N_train <- 10
X <- runif(n = N_train, min = -15, max = 10)
Y <- fgx(X) + 4*rnorm(length(X))
```

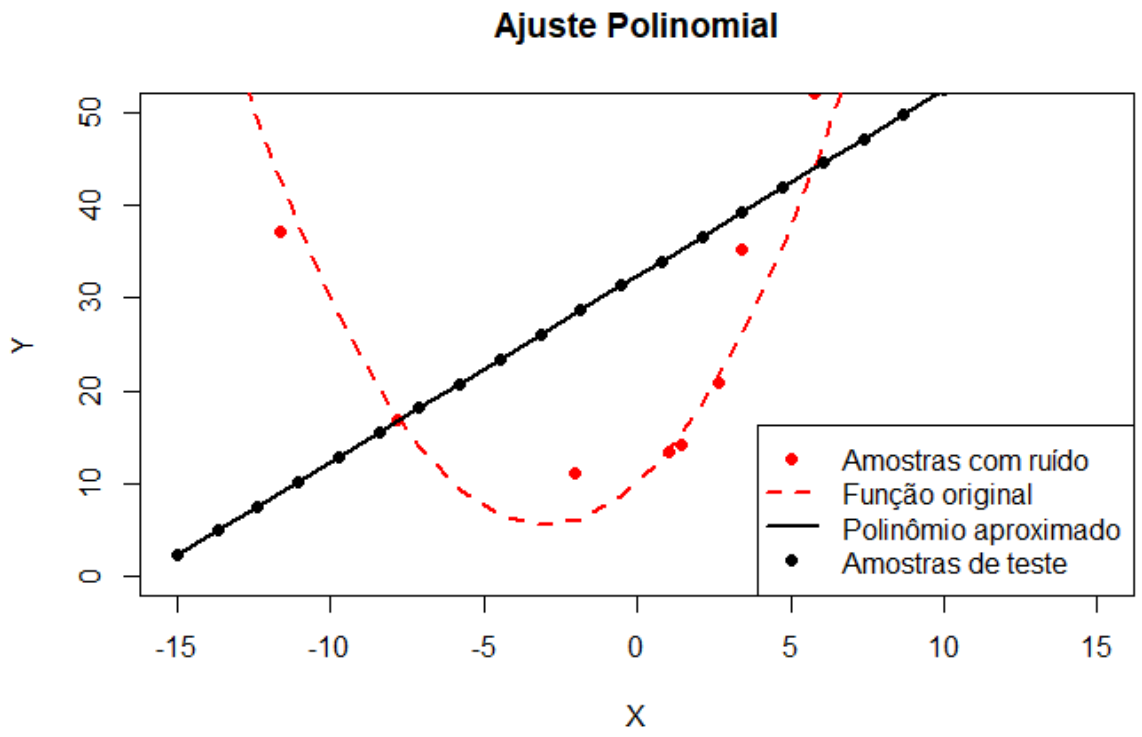
Em seguida, cria-se a matriz de features para o ajuste polinomial, inicialmente com grau 1, bem como define-se o vetor de teste e xt com 20 amostras assim como requisitado no enunciado.

```
# Criando matriz de características para ajuste polinomial
p <- 1
H <- cbind(X^p, X, 1)
w <- pseudoinverse(H) %*% Y  # obtendo os coeficientes

#Criar 20 pontos para avaliar a aproximação
N_eval <- 20
xt <- seq(-15, 10, length.out = N_eval)
Ht <- cbind(xt^p, xt, 1)
Yhattst <- Ht %*% w  # Valores previstos pelo modelo ajustado
```

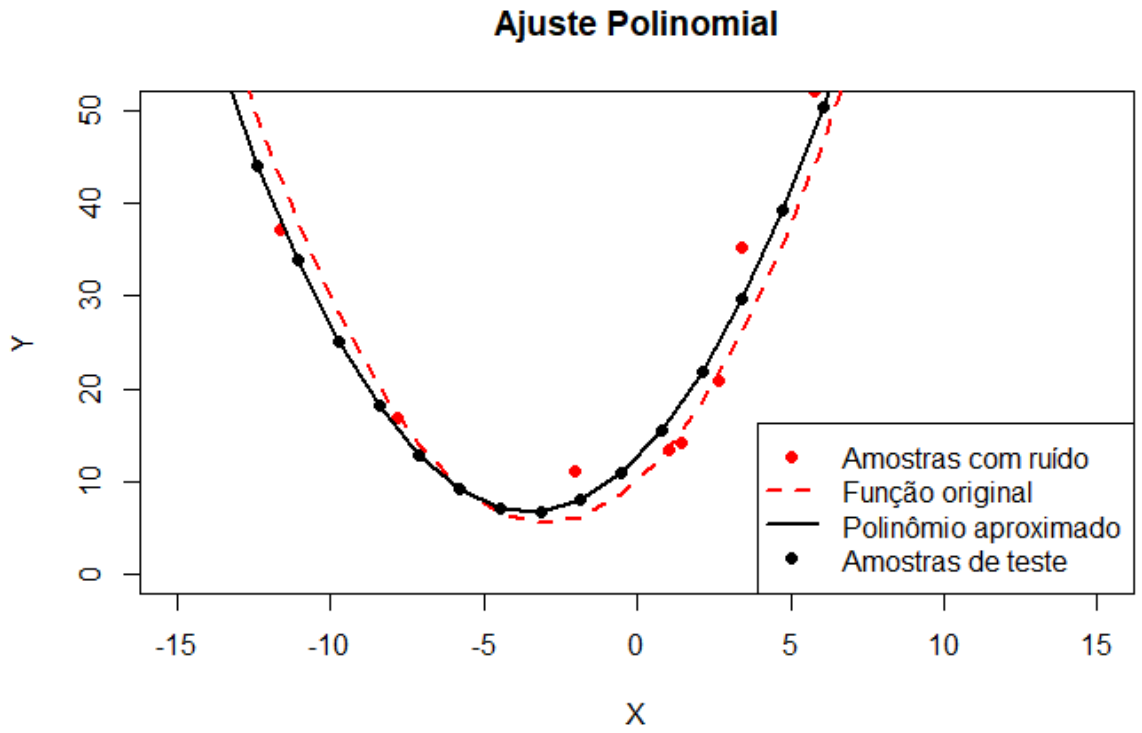
Por fim, plotam-se a função geradora e os 10 pontos referentes às amostras (em cor vermelha),bem como a função aproximada pelo modelo e os pontos de teste (em cor preta).

```
# Plotagem
plot(X, Y, col = 'red', pch = 16, xlab = "X", ylab = "Y",
     main = "Ajuste Polinomial", xlim = c(-15, 15), ylim = c(0, 50))

lines(xt, yg, col = 'red', lwd = 2, lty = 2) # Função original
lines(xt, Yhattst, col = 'black', lwd = 2) # Aproximação polinomial
points(xt, Yhattst, col = "black", pch = 16) # Adiciona os 20 pontos de teste

legend("bottomright",
       legend = c("Amostras com ruído", "Função original",
                  "Polinômio aproximado", "Amostras de teste"),
       col = c("red", "red", "black", "black"),
       pch = c(16, NA, NA, 16), lty = c(NA, 2, 1, NA), lwd = c(NA, 2, 2, NA))
```

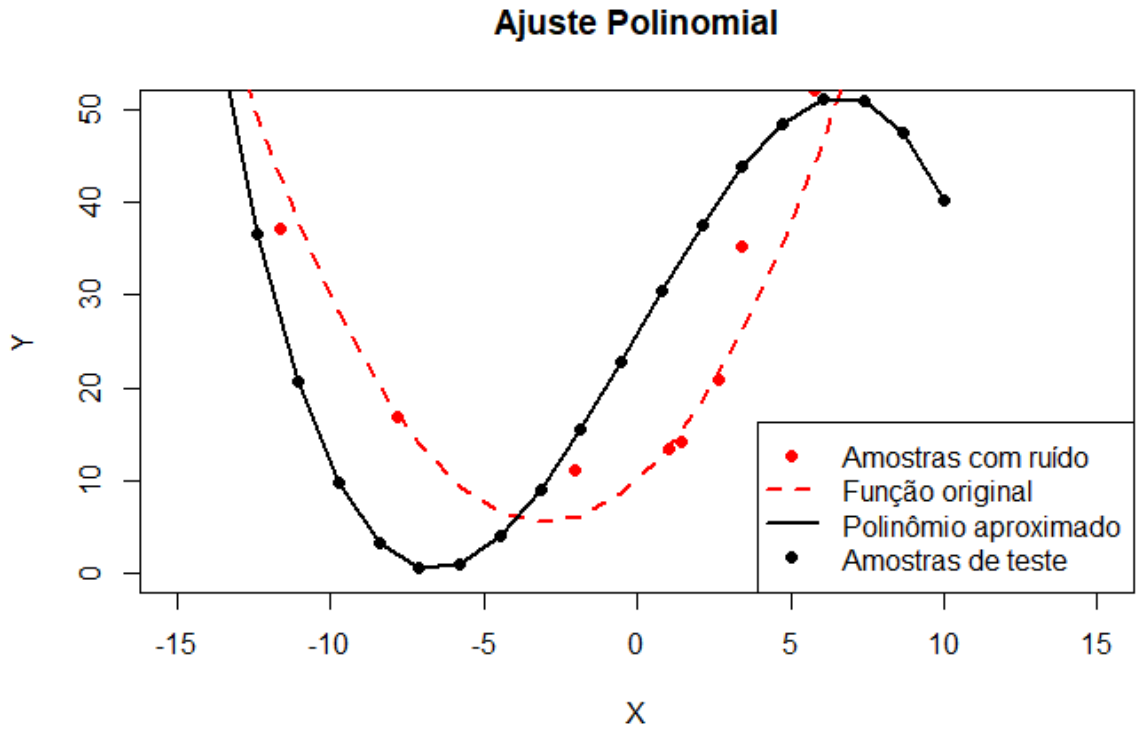
Agora, varia-se o grau (p) do polinômio aproximado de 1 à 8, obtendo os seguintes gráficos:



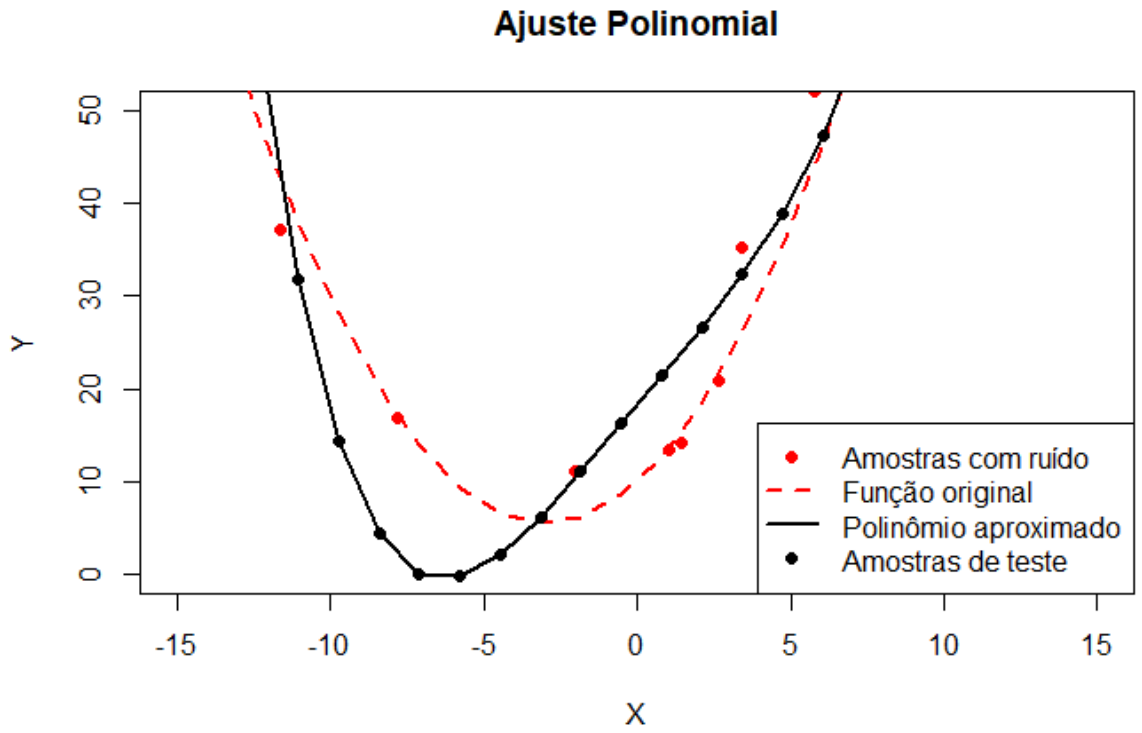

p = 1


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

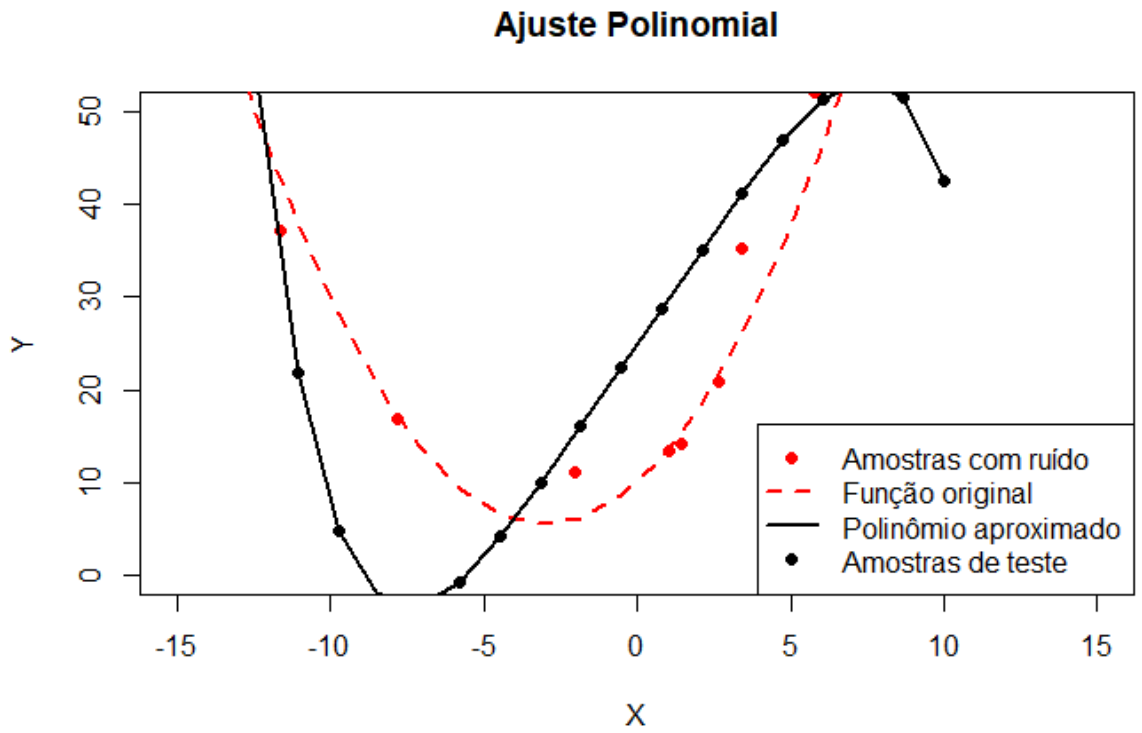
p = 2


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

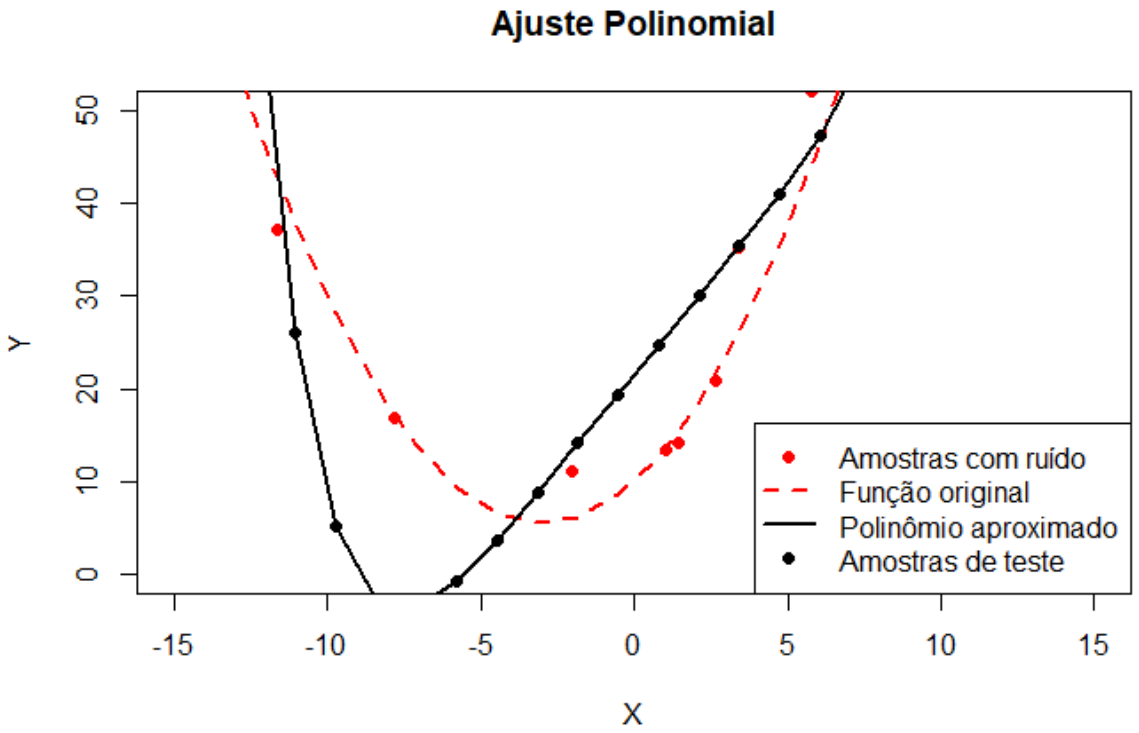
p = 3


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

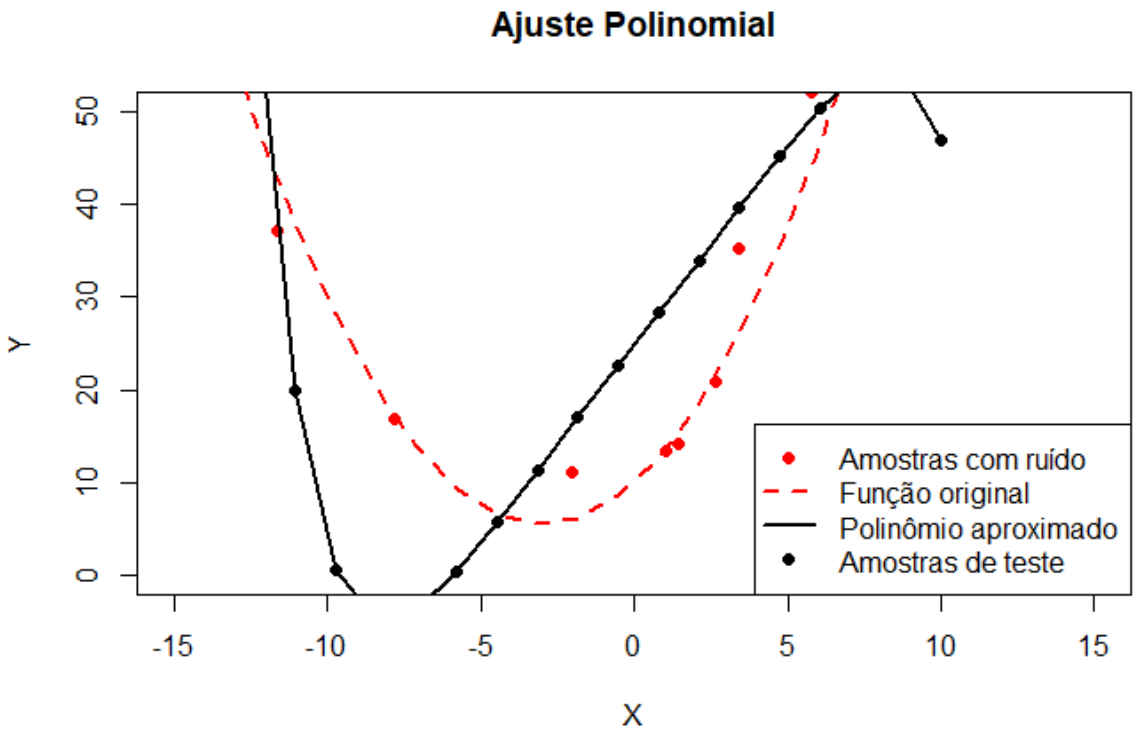
p = 4


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

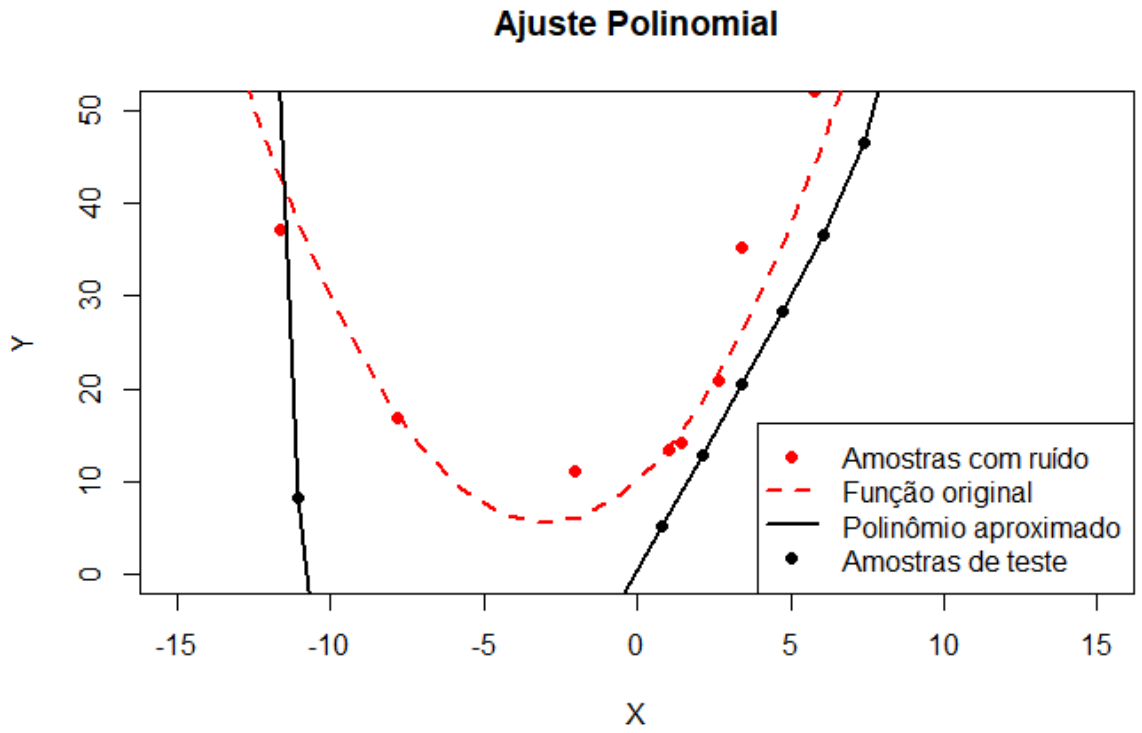
p = 5


Ajuste Polinomial
Y
X
-15
-10
-5
0
5
10
15
0
10
20
30
40
50
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

p = 6


Ajuste Polinomial
Y
X
-15
-10
-5
0
5
10
15
0
10
20
30
40
50
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

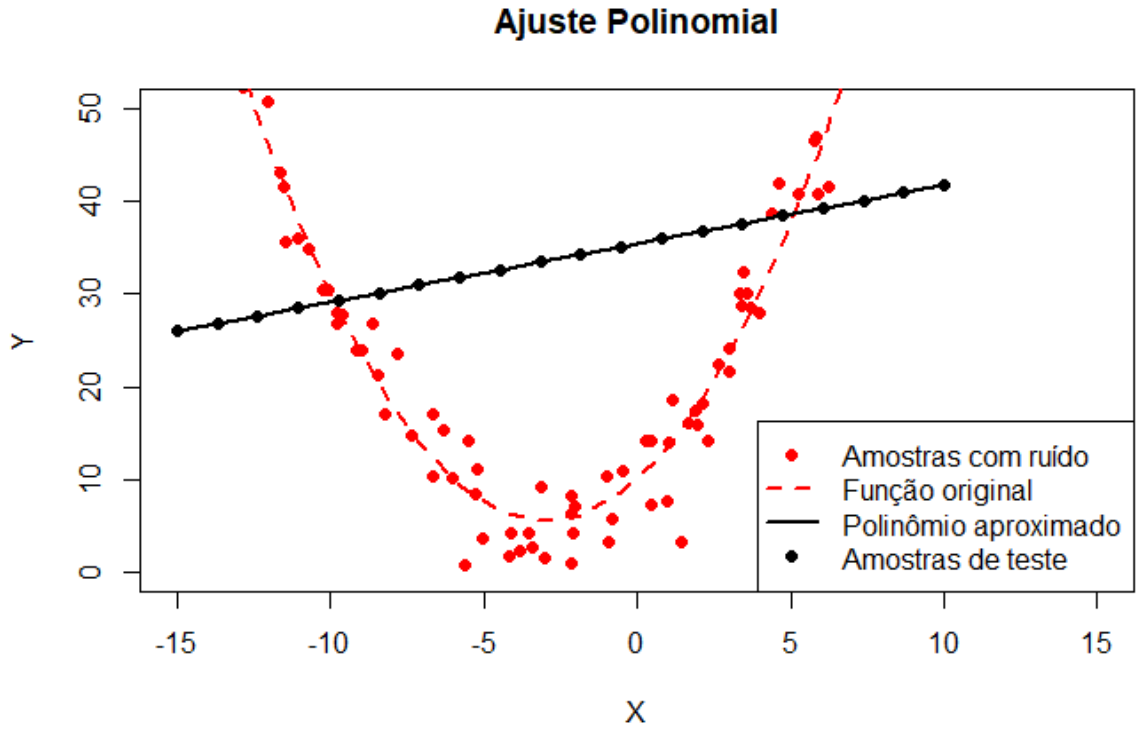
p = 7



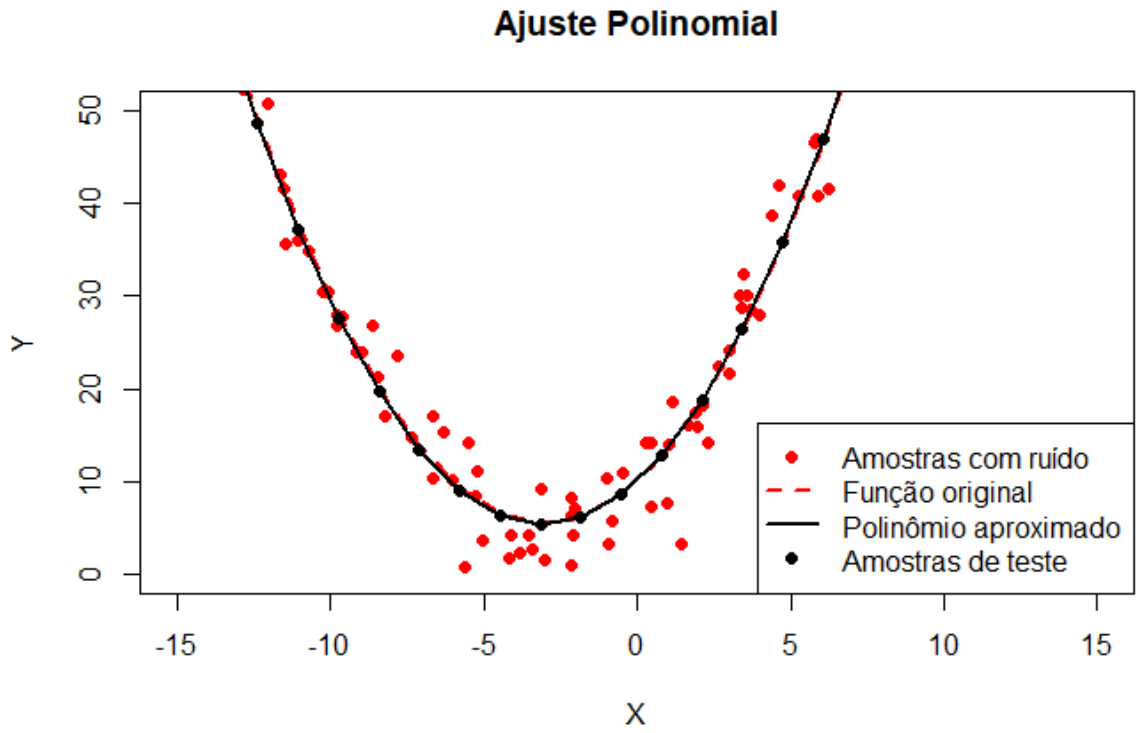

p = 8

A melhor função aproximada obviamente corresponde ao polinômio de grau 2, já que a função geradora também possui grau 2. Diferentemente do esperado, não parece haver overfitting para graus maiores da função aproximada - já que o erro de treinamento parece se manter alto para qualquer grau diferente do ideal -, mas sim um forte desvio com relação à função geradora.

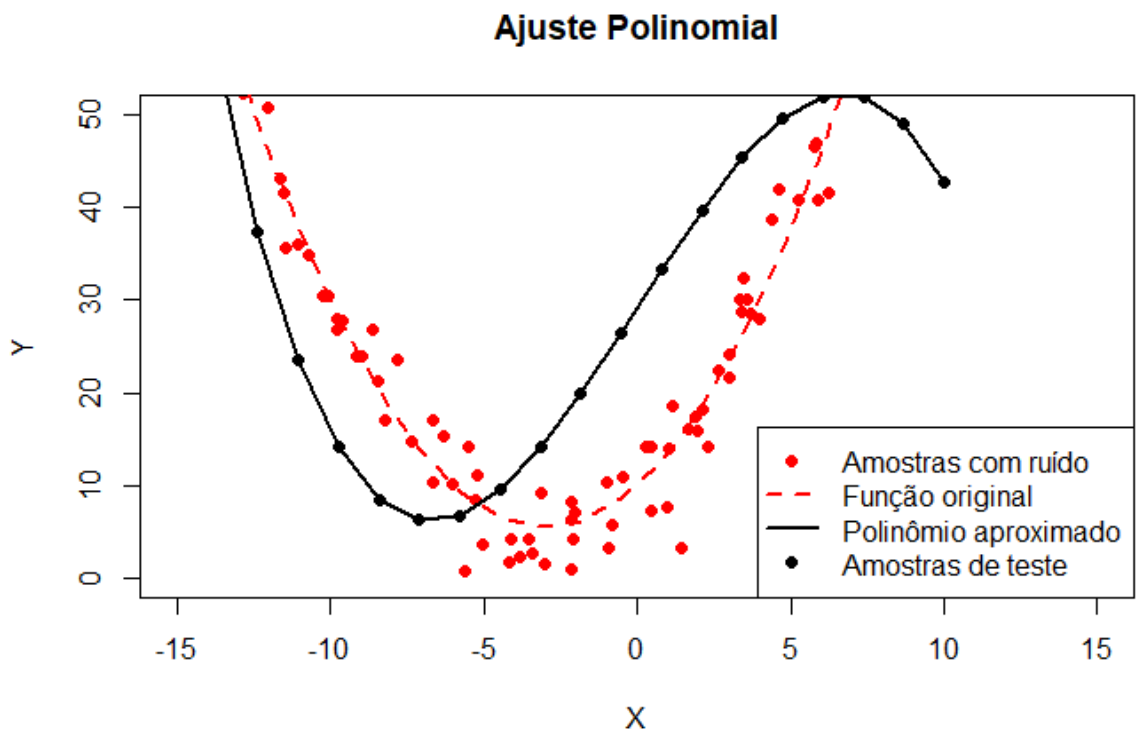
Por fim, o procedimento foi repetido com 100 amostras de treinamento.



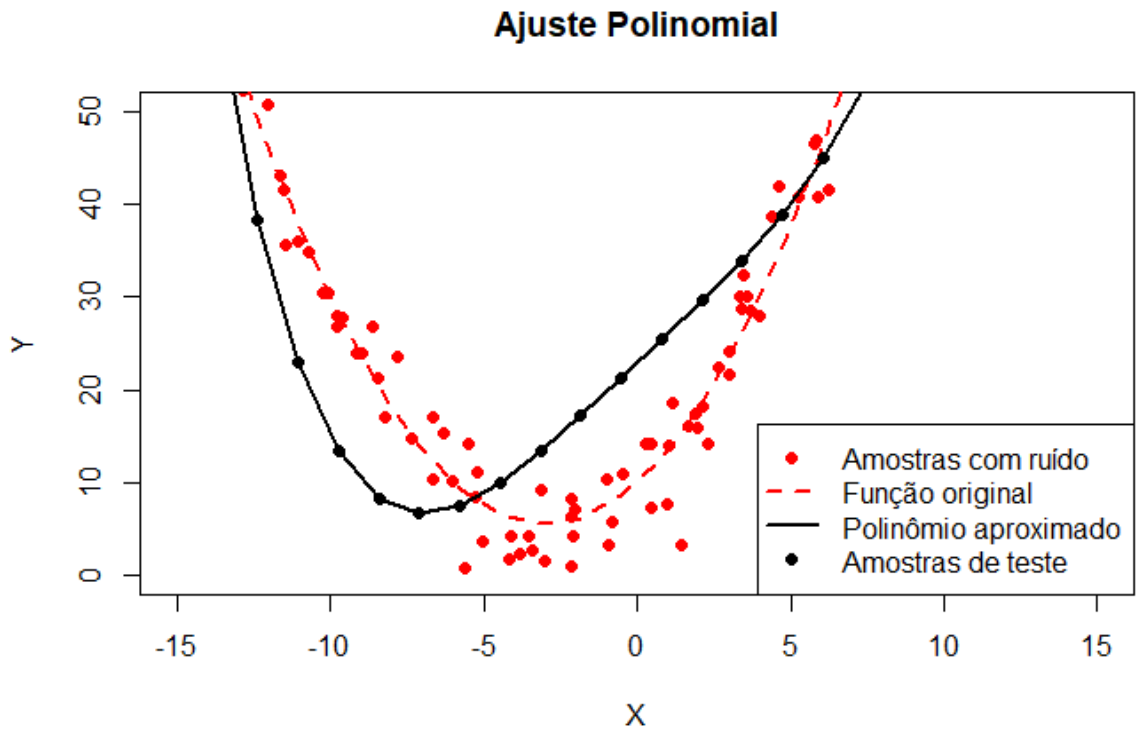

p = 1



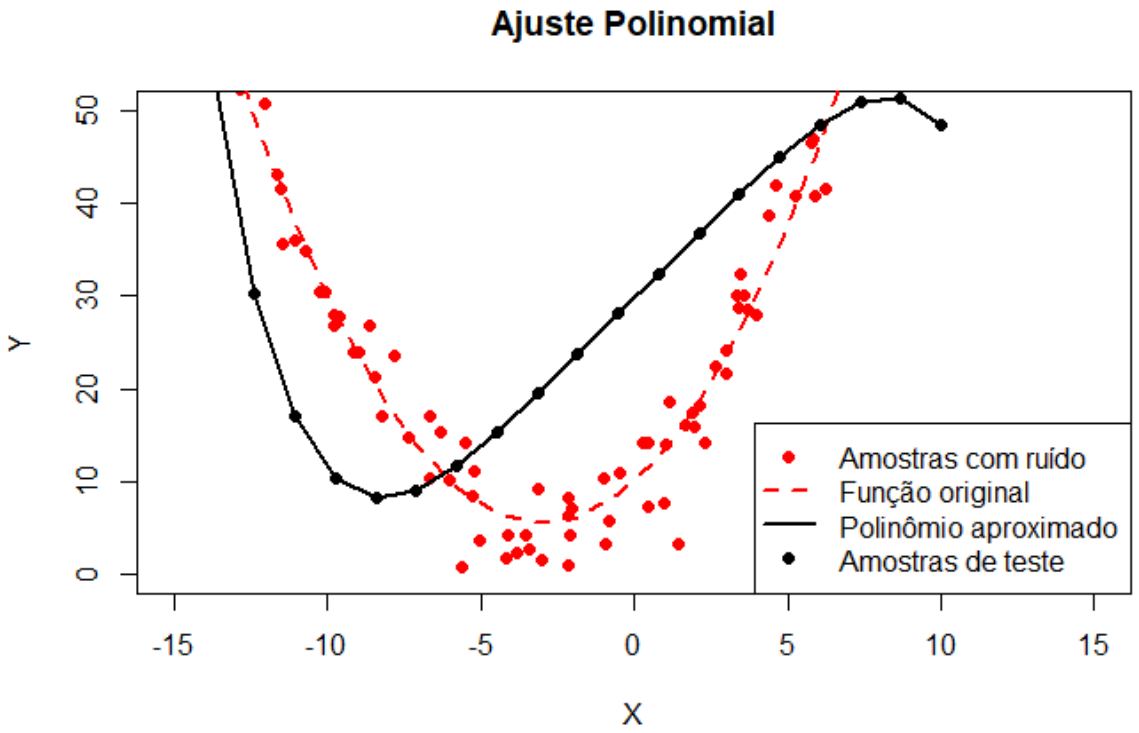

p = 2


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

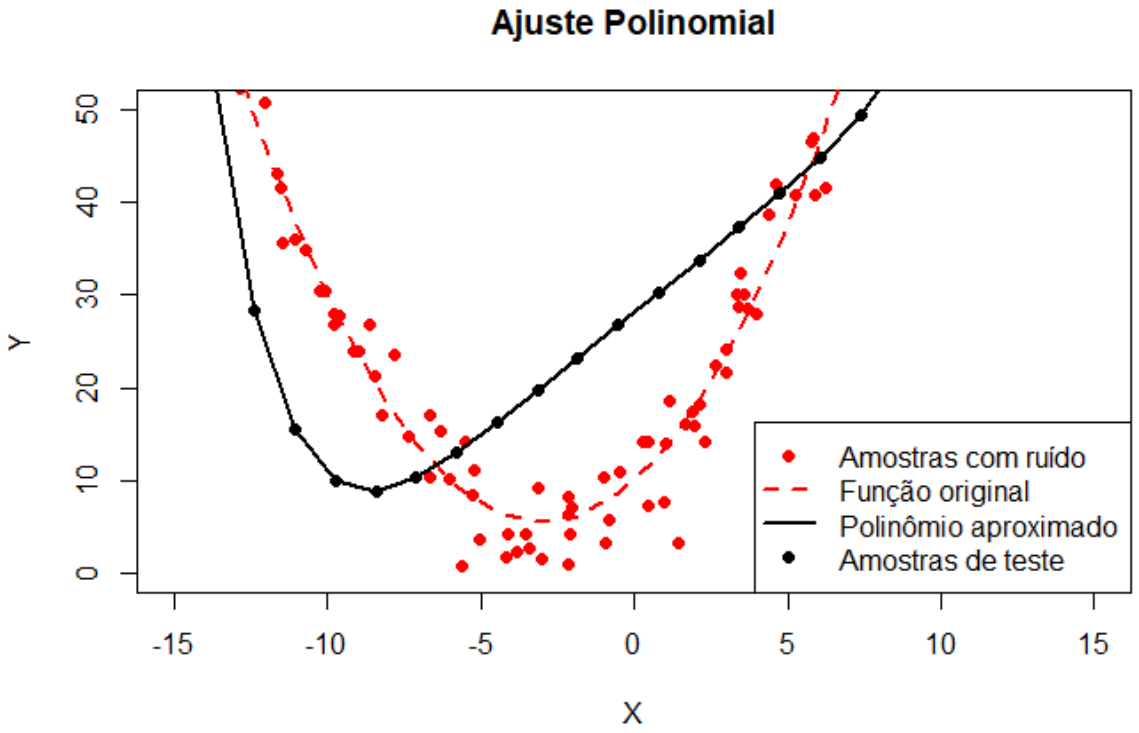
p = 3


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

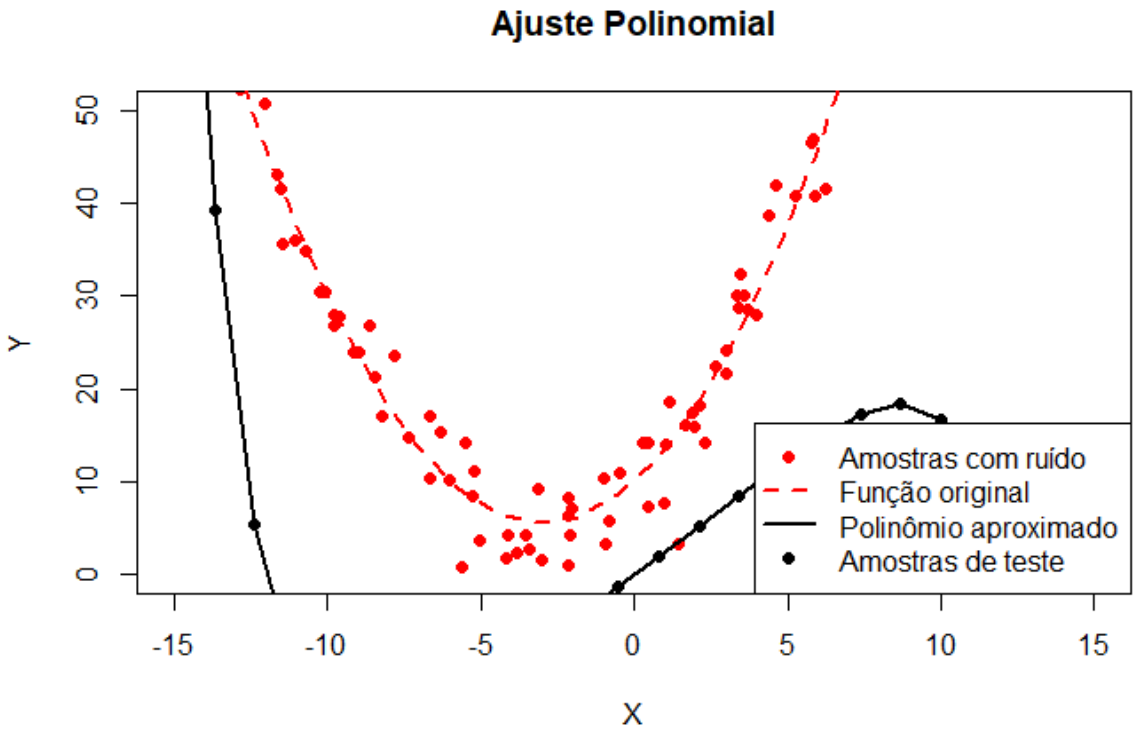
p = 4


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

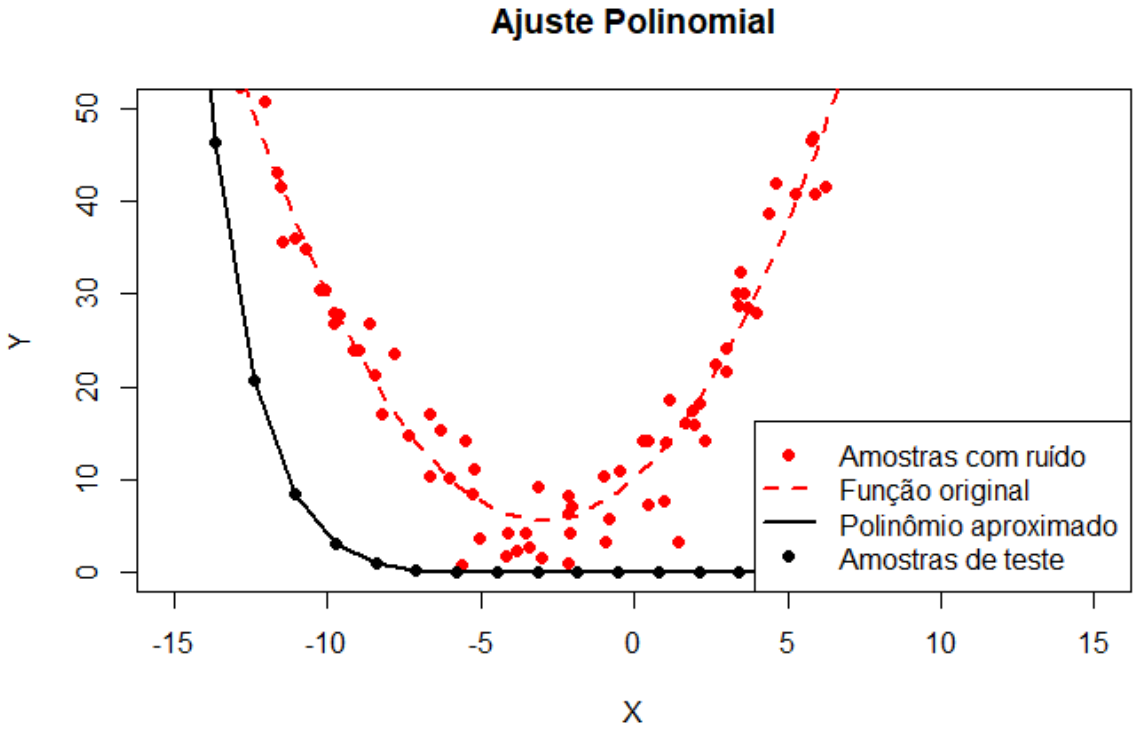
p = 5


Ajuste Polinomial
Y
X
0
10
20
30
40
50
-15
-10
-5
0
5
10
15
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

p = 6


Ajuste Polinomial
Y
X
-15
-10
-5
0
5
10
15
0
10
20
30
40
50
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

p = 7


Ajuste Polinomial
Y
X
-15
-10
-5
0
5
10
15
0
10
20
30
40
50
Amostras com ruído
Função original
Polinômio aproximado
Amostras de teste

p = 8

A utilização de 100 amostras de treino parece resultar em resultados com menor erro de treinamento em relação aos resultados obtidos com 10 amostras, não apenas isto, mas funções aproximadas mais próximas da função geradora em geral para cada grau testado. Por exemplo, a função aproximada com grau e 100 amostras de teste aparenta possuir um erro bem menor em relação à função geradora do que o obtido pela função aproximada com mesmo grau e apenas 10 amostras.

No entanto, assim como nos testes anteriores, embora o polinômio aproximado de grau 1 aparente um underfitting, funções de maior grau não parecem apresentar overfitting, já que o erro de treinamento permanece alto.