Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Estatística

Mineração de dados

Adriana Eva Fernandes da Silva

1. Inicialmente, o objetivo é avaliar qual das 13 covariáveis contínuas se relaciona melhor com a variável resposta preço. Notamos que, ao se observar a Tabela 1, de acordo com o Teste de Shapiro-Wilk, nenhuma das covariáveis em questão são providas de uma distribuição normal. Desse modo, é arriscado se basear na correlação de Pearson, visto que esta depende da suposição de normalidade para ambas as variáveis. Assim, a escolha estará pautada na correlação não-paramétrica de *Spearman* e nas tendências da Figura 1. O maior valor da correlação refere-se à covariável *curbweight*, que está relacionada, digamos, ao peso real do carro. Todavia, observamos que a covariável *horsepower* tem uma correlação muito próxima e, na prática, faz mais sentido sua escolha, já que carros mais potentes tendem a custar mais caro. Também, a tendência no gráfico de dispersão entre o preço e a potência do carro é de aproximadamente uma reta, reforçando a escolha. Portanto, a covariável *horsepower* será utilizada para predizermos o preço do carro.

Tabela 1: Correlação de *Pearson* e *Speraman* entre o preço e as 13 covariáveis contínuas presentes no banco de dados, e o p-valor do teste de *Shapiro-Wilk*

Covariáveis	$cor_{Pearson}$	$cor_{Spearman}$	*Shapiro-Wilk*
wheelbase	0.58	0.68	< 0.0001
carlength	0.68	0.80	0.0104
carwidth	0.76	0.81	< 0.0001
carheight	0.12	0.24	0.0217
curbweight	0.84	0.91	< 0.0001
enginesize	0.87	0.83	< 0.0001
boreratio	0.55	0.64	1e-04
stroke	0.08	0.11	< 0.0001
compressionratio	0.07	-0.17	< 0.0001
horsepower	0.81	0.85	< 0.0001
peakrpm	-0.09	-0.07	3e-04
citympg	-0.69	-0.83	< 0.0001
highwaympg	-0.70	-0.82	7e-04

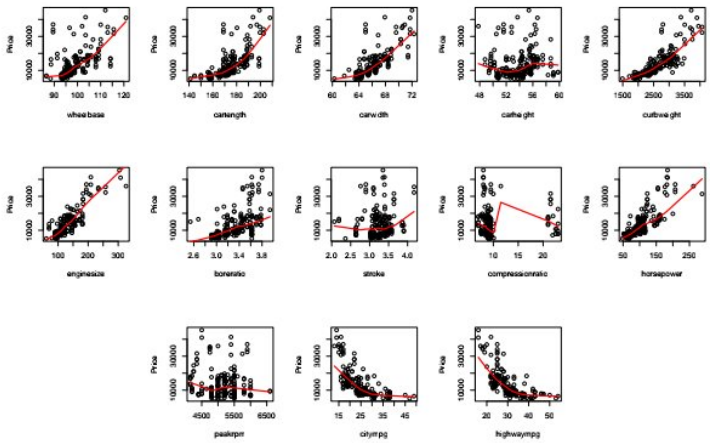

Figura 1: Gráfico de dispersão entre o preço e as 13 covariáveis contínuas disponíveis no banco de dados

2. Utilizando-se o método dos mínimos quadrados e a validação cruzada *leave-one-out*, foram obtidos os seguintes riscos para cada um dos 30 polinômios:

Tabela 2: Risco da regressão para $p = 1, \ldots, 30$

Risco	Ordem da regressão polinomial p
64773199.86	1
65276500.91	2
63813543.54	3
63715024.10	4
63976828.73	5
64760822.16	6
64864441.25	7
68438667.51	8
68435665.51	9
67533204.50	10
67533204.50	11
66215325.81	12
66215325.81	13
64741020.80	14
64741020.80	15
64741020.80	16
64741020.80	17
65137546.39	18
65137546.39	19
65137546.39	20
65137546.39	21
65137546.39	22
65137546.39	23
65301188.19	24
65301188.19	25
65301188.19	26
65301188.19	27
65301188.19	28
65301188.19	29
65301188.19	30

3. Ao observarmos a Figura 2, verificamos que, segundo o critério do risco da regressão, para o valor $p = 4$ tal quantidade é a menor. Portanto, $p_{esc} = 4$.

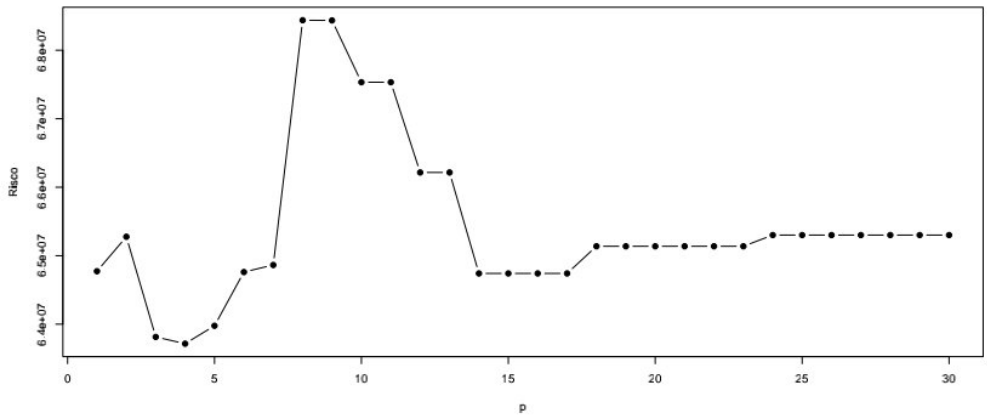

Figura 2: Gráfico do risco vs p

4. A Figura 3 nos mostra que, aparentemente, não há muita diferença entre os ajustes de $p = 1$ e $p = 4$ visto que eles, de certa forma, estão acompanhando a tendência dos dados. Nesse caso, no contexto da regressão polinomial, o que geralmente é considerado um *sub-ajuste* ($p = 1$) está se equiparando com o ajuste intermediário ($p = 4$). Para $p = 30$, tem-se um *super-ajuste*, visto que a curva tenta se adequar localmente aos dados, o que de certa forma é um exagero. Assim, pela sua alta complexidade, o ajuste referente a $p = 30$ será desconsiderado. Finalmente, como já foi frisado, por não haver muita diferença no desempenho preditivo entre as duas primeiras ordens, iremos optar pelo ajuste mais simples, isto é, $p = 1$.

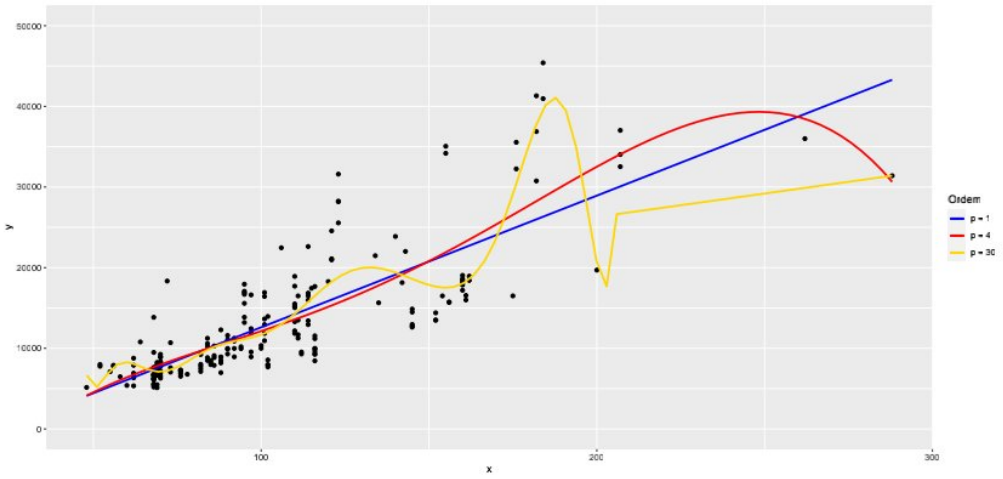

Figura 3: Gráfico de y (preço) vs x (potência do motor) com as curvas da regressão polinomial para $p = 1, 4, 30$ sobrepostas

5. Por intermédio da Figura 4, notamos que o ajuste para $p = 30$ está produzindo um resultado bastante estranho. Simplesmente, um dos valores preditos está negativo, o que pode ser considerado um absurdo. Quanto aos outros ajustes, bons resultados foram obtidos, e são bastante semelhantes. Assim sendo, optar por um modelo mais complexo não faz muito sentido. Desse modo, novamente, será escolhido o modelo mais simples, $p = 1$.

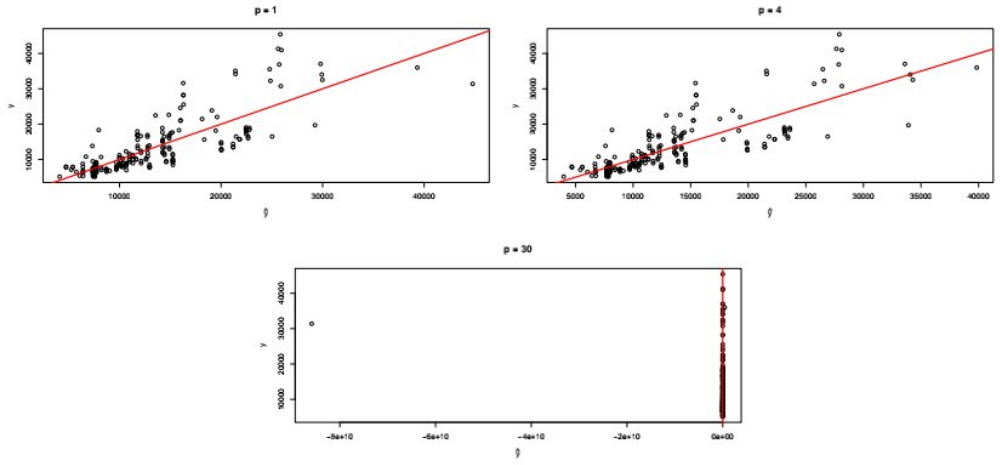

Figura 4: Valores preditos vs y para a regressão polinomial de ordens $p = 1$, $p = 4$ e $p = 30$

6. A vantagem do método *leave-one-out* é o fato de que, em razão de se ajustar repetidamente um modelo a um conjunto de dados que contém $n-1$ observações, tende a não superestimar o teste do *MSE* em comparação à utilização de uma única amostra de teste. Em compensação, pode levar a um processo bastante demorado quando n é grande, com o possível agravante do modelo ser muito complexo e assim, consumir ainda mais tempo.

Códigos no R

```
################ Lista de Mineração

require(xtable)
require(ggplot2)

setwd("C:\\Users\\User\\Downloads")

dados = read.csv("atividade-1-precificacao-veiculos.csv", h = T)


######## Item 1

var_cont = dados[, c("wheelbase", "carlength", "carwidth", "carheight",
                     "curbweight", "enginesize", "boreratio", "stroke",
                     "compressionratio", "horsepower", "peakrpm", "citympg",
                     "highwaympg")]

layout(matrix(c(0, 1, 2, 3, 4, 5, 0, 0, 6, 7, 8, 9, 10, 0, 0, 0, 11, 12, 13,
0, 0), nrow = 3, ncol = 7, byrow = T))

cor_pearson = c()
cor_spearman = c()
shapiro.var = c()

for(i in 1:13)
{
 plot(var_cont[, i], dados$price, xlab = colnames(var_cont)[i], ylab = "Price")
 lines(lowess(var_cont[, i], dados$price), col = "red", lwd = 2)
 cor_pearson[i] = cor(dados$price, var_cont[, i], method = c("pearson"))
 cor_spearman[i] = cor(dados$price, var_cont[, i], method = c("spearman"))
 shapiro.var[i] = shapiro.test(var_cont[, i])$p
```

```
}

tabela = data.frame(colnames(var_cont),
                    round(cor_pearson, 2), round(cor_spearman, 2),
                    shapiro.var)
tabela[, 4] = ifelse(abs(tabela[, 4]) < 0.0001, "< 0.0001",
                     round(tabela[, 4], 4))
colnames(tabela) = c("Covariáveis", "cor_pearson", "cor_spearman", "pvalor")

xtable(tabela)

######## Item 2

y = dados$price
x = dados$horsepower

p = 30
n = nrow(dados)
I_pol = paste("I(x^", 1:p, ")", sep = "")
polinomios = paste("y ~", sapply(1:p, function(i){paste(I_pol[1:i],
                                                        collapse = "+")}))
mse = c()
risco = c()

for(i in 1:p)
{
 for(j in 1:n)
 {
  y_treinamento = y[-j]
  x_treinamento = x[-j]
  modelo = lm(formula(polinomios[i]))
  predito = predict(modelo, newdata = list(y_treinamento = y[j],
                                           x_treinamento = x[j]))
  mse[j] = (y[j] - predito)^2
}

risco[i] = mean(mse)
}

tabela2 = cbind(risco, 1:p)
colnames(tabela2) = c("Risco", "Ordem da regressão polinomial")
```

```
xtable(tabela2)


######## Item 3

par(mfrow = c(1, 1))

plot(1:30, risco, ylab = "Risco", xlab = "p", type = "b", pch = 19)


######## Item 4


ggp = ggplot(data.frame(x, y), aes(x, y)) + geom_point() + ylim(0, 50000)

ggp +  geom_smooth(method = "lm", se = FALSE, aes(color = "blue") ) +
       geom_smooth(method = "lm", formula = y ~ poly(x, 4, raw = T),
       se = FALSE, aes(color = "red") ) +
       geom_smooth(method = "lm", formula = y ~ poly(x, 30, raw = T),
       se = FALSE, aes(color = "gold") ) +
       scale_color_identity(name = "Ordem",
                            breaks = c("blue", "red", "gold"),
                            labels = c("p = 1", "p = 4", "p = 30"),
                            guide = "legend")


######## Item 5

p = c(1, 4, 30)

preditos = matrix(NA, nrow = nrow(dados), ncol = length(p))

for(i in 1:length(p))
{
 for(j in 1:n)
 {
  y_treinamento = y[-j]
  x_treinamento = x[-j]
  modelo = lm(y_treinamento ~ poly(x_treinamento, degree = p[i], raw = T))
  preditos[j, i] = predict(modelo, newdata =
                                  list(y_treinamento = y[j],
```

```
                                    x_treinamento = x[j]))
 }
}

layout(matrix(c(1, 1, 2, 2, 0, 3, 3, 0), nrow = 2, ncol = 4, byrow = T))
plot(preditos[, 1], y, xlab = expression(hat(y)), main = "p = 1")
abline(coef = c(0,1), col = "red", lwd = 2)
plot(preditos[, 2], y, xlab = expression(hat(y)), main = "p = 4")
abline(coef = c(0,1), col = "red", lwd = 2)
plot(preditos[, 3], y, xlab = expression(hat(y)), main = "p = 30")
abline(coef = c(0,1), col = "red", lwd = 2)
```