R Notebook

# Prova de Especialização

* Primeiramente vamos importar os dados:

wine.data <- readxl::read\_excel('exe.xlsx', 'D')  
head(wine.data)

## # A tibble: 6 x 6  
## Clari Aroma Corpo Sabor Afina Qua  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 3.3 2.8 3.1 4.1 9.8  
## 2 1 4.4 4.9 3.5 3.9 12.6  
## 3 1 3.9 5.3 4.8 4.7 11.9  
## 4 1 3.9 2.6 3.1 3.6 11.1  
## 5 1 5.6 5.1 5.5 5.1 13.3  
## 6 1 4.6 4.7 5 4.1 12.8

Os dados do vinho são referentes aos atributos Claridade, Aroma, Corpo, Sabor, Afinação e Qualidade. O nosso objetivo no estudo é relacionar os atributos Claridades, Aroma, Corpo, Sabor e Afinação com a Qualidade do vinho, ou seja, quais desses atributos interferem na qualidade.

#### **a) Estime pelo método dos mínimos quadrados. Explique o procedimento.**

O objetivo é saber os valores , , , , que minimize a soma…

Para isso, precisa-se calcular a *Qualidade Estimada* (), que é:

Claridade <- wine.data$Clari  
Aroma <- wine.data$Aroma  
Corpo <- wine.data$Corpo  
Sabor <- wine.data$Sabor  
Afinacao <- wine.data$Afina  
Y <- wine.data$Qua  
  
desvioY <- function(betas) {  
 Y.Estimado <- betas[1] + betas[2]\*Claridade + betas[3]\*Aroma + betas[4]\*Corpo + betas[5]\*Sabor + betas[6]\*Afinacao  
 S <- sum( (Y - Y.Estimado)^2 )  
}

A função desvioY, calcula o resíduo dos Y (), eleva ao quadrado para não haver números negativos eliminando números positivos e por fim soma tudo. É essa função que se gostaria que fosse a menor possível.

Ou seja, o objetivo é saber os valores dos , , , , que minimize a soma S.

Aplicando a função optim temos os betas procurados:

R.2 <- optim(par = c(1,1,1,1,1,1), fn = desvioY, method = "L-BFGS-B")  
R.2$par

## [1] 3.9962004 2.3384721 0.4826345 0.2732280 1.1682793 -0.6837678

Executando a regressão linear para ter valores matemáticos mais exatos, temos:

U <- lm(Y~Claridade+Aroma+Corpo+Sabor+Afinacao)  
U$coefficients

## (Intercept) Claridade Aroma Corpo Sabor Afinacao   
## 3.9968648 2.3394535 0.4825505 0.2731612 1.1683238 -0.6840102

Comparando os valores dos pelo método dos mínimos quadrados com os valores reais da regressão linear, observamos que os valores estão bem próximos.

#### **b) Você concorda que uma relação linear é adequada? Como avaliaria no caso da Regressão Linear Múltipla?**

Caso tivéssemos a variável de resposta Y relacionada com uma única variável independente X, seria suficiente fazer um diagrama de dispersão do relacionando Y com X. Porém, nós temos 5 variáveis (dimensões), sem adicionar o Y.

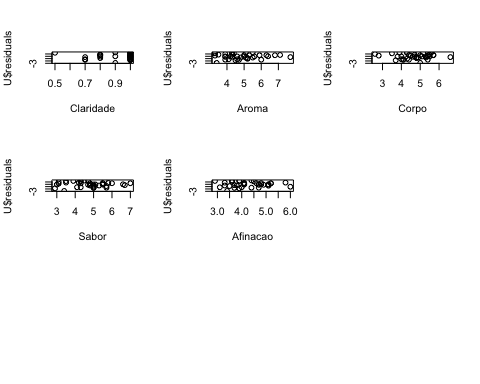
Portanto, precisamos fazer um gráfico de dispersão do resíduo com cada uma das variáveis independentes (Claridade, Aroma, Corpo, Sabor, Afinacao), esperando um padrão aleatório em cada um desses gráficos.

Lembrando que *Resídio* é tudo que não foi explicado da qualidade do vinho.

Plotando os gráficos relacionando as variávies independentes (Claridade, Aroma, Corpo, Sabor, Afinacao) em relação com o Resíduo *não* se observa um padrão linear em nenhuma delas.

Logo, é razoável se trabalhar com uma Regressão Linear, pois o que se sobrou no resíduo não possui efeito não linear no modelo.

par(mfrow=c(3,3))  
  
plot(Claridade, U$residuals)  
plot(Aroma, U$residuals)  
plot(Corpo, U$residuals)  
plot(Sabor, U$residuals)  
plot(Afinacao, U$residuals)

 #### **c)Estime a variância da componente erro utilizando um estimador não viciado (). Explique o procedimento**

A variância não viciada do componente erro trata-se da variância dos resíduos corrigida pelo número de parâmetro estimados, logo:

$^2 = Var(residuos) / (n - Quantidade De Parâmetros Estimados) $

*Observação: utilizamos do artifício matemático de multiplicar por (n -1) para eliminar o denominador da fórmula utilizada no cálculo da variância dos resíduos.*

n <- 38  
quantidadeParametrosEstimados <- 6  
sigma.2 <- var(U$residuals) \* (n -1 ) / (n - quantidadeParametrosEstimados)  
sigma.2

## [1] 1.3515

#### **d) Estime a variância de (relativo ao sabor). Explique o procedimento.**

Iremos estimar a variância através da matriz utilizando a estimativa de mínimos quadrados de

* Para obtermos a Matriz de planejamento X:

X <- model.matrix(U)  
head.matrix(X)

## (Intercept) Claridade Aroma Corpo Sabor Afinacao  
## 1 1 1 3.3 2.8 3.1 4.1  
## 2 1 1 4.4 4.9 3.5 3.9  
## 3 1 1 3.9 5.3 4.8 4.7  
## 4 1 1 3.9 2.6 3.1 3.6  
## 5 1 1 5.6 5.1 5.5 5.1  
## 6 1 1 4.6 4.7 5.0 4.1

* Calculando a transposta da matriz:

X.linha <- t(X)  
head.matrix(X.linha)

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
## (Intercept) 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  
## Claridade 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.5 0.8 0.7 1.0 0.9 1.0  
## Aroma 3.3 4.4 3.9 3.9 5.6 4.6 4.8 5.3 4.3 4.3 5.1 3.3 5.9 7.7 7.1 5.5 6.3  
## Corpo 2.8 4.9 5.3 2.6 5.1 4.7 4.8 4.5 4.3 3.9 4.3 5.4 5.7 6.6 4.4 5.6 5.4  
## Sabor 3.1 3.5 4.8 3.1 5.5 5.0 4.8 4.3 3.9 4.7 4.5 4.3 7.0 6.7 5.8 5.6 4.8  
## Afinacao 4.1 3.9 4.7 3.6 5.1 4.1 3.3 5.2 2.9 3.9 3.6 3.6 4.1 3.7 4.1 4.4 4.6  
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34  
## (Intercept) 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  
## Claridade 1.0 1.0 0.9 0.9 1.0 0.7 0.7 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.8 1.0 1.0  
## Aroma 5.0 4.6 3.4 6.4 5.5 4.7 4.1 6.0 4.3 3.9 5.1 3.9 4.5 5.2 4.2 3.3 6.8  
## Corpo 5.5 4.1 5.0 5.4 5.3 4.1 4.0 5.4 4.6 4.0 4.9 4.4 3.7 4.3 3.8 3.5 5.0  
## Sabor 5.5 4.3 3.4 6.6 5.3 5.0 4.1 5.7 4.7 5.1 5.0 5.0 2.9 5.0 3.0 4.3 6.0  
## Afinacao 4.1 3.1 3.4 4.8 3.8 3.7 4.0 4.7 4.9 5.1 5.1 4.4 3.9 6.0 4.7 4.5 5.2  
## 35 36 37 38  
## (Intercept) 1.0 1.0 1.0 1.0  
## Claridade 0.8 0.8 0.8 0.8  
## Aroma 5.0 3.5 4.3 5.2  
## Corpo 5.7 4.7 5.5 4.8  
## Sabor 5.5 4.2 3.5 5.7  
## Afinacao 4.8 3.3 5.8 3.5

* Multiplica a tranposta (X’) por X:

X1 <- X.linha%\*%X  
head.matrix(X1)

## (Intercept) Claridade Aroma Corpo Sabor Afinacao  
## (Intercept) 38.0 35.10 184.20 178.00 181.20 161.70  
## Claridade 35.1 32.99 170.45 163.25 166.97 149.98  
## Aroma 184.2 170.45 936.24 880.95 908.67 789.78  
## Corpo 178.0 163.25 880.95 858.92 869.05 760.86  
## Sabor 181.2 166.97 908.67 869.05 903.14 776.10  
## Afinacao 161.7 149.98 789.78 760.86 776.10 708.23

* Calculando a inversa de :

X.inversa <- solve(X1)  
head.matrix(X.inversa)

## (Intercept) Claridade Aroma Corpo Sabor  
## (Intercept) 3.68538435 -2.16087923 0.056193065 -0.29618518 -0.02052109  
## Claridade -2.16087923 2.22687731 -0.081481738 0.16333471 0.01106341  
## Aroma 0.05619307 -0.08148174 0.054922315 -0.01420775 -0.03576244  
## Corpo -0.29618518 0.16333471 -0.014207755 0.08183244 -0.02875529  
## Sabor -0.02052109 0.01106341 -0.035762437 -0.02875529 0.06859675  
## Afinacao -0.10580781 -0.07494938 -0.002367866 -0.00752418 -0.00205549  
## Afinacao  
## (Intercept) -0.105807813  
## Claridade -0.074949380  
## Aroma -0.002367866  
## Corpo -0.007524180  
## Sabor -0.002055490  
## Afinacao 0.054417686

* Calculando a variância do modelo:

U.summary <- summary(U)  
variancia.modelo <- (U.summary$sigma)^2  
variancia.modelo

## [1] 1.3515

* Por fim, calculando a variância de . Observamos na matriz *X.inversa* (calculada acima) que o valor da estimativa do mínimo quadrado de (sabor) encontra-se na linha 5 e coluna 5.

beta4.variancia <- variancia.modelo \* X.inversa[5,5]  
beta4.variancia

## [1] 0.09270852