Checkpoint 2 - Nível do PSA

Leonardo Alves dos Santos

April 1, 2016

```
#Libraries
library(plyr)
library(dplyr)
library(ggplot2)
library(caret)
```

```
#http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/prostate.data
prostate.data <- read.delim("./data/prostate.data.txt")
```

Introdução

Os dados utilizados representam resultados de exames realizados em pacientes do sexo Masculino, com o objetivo de diagnosticar pacientes com Câncer de Prostata. O dataset possui os seguintes dados.

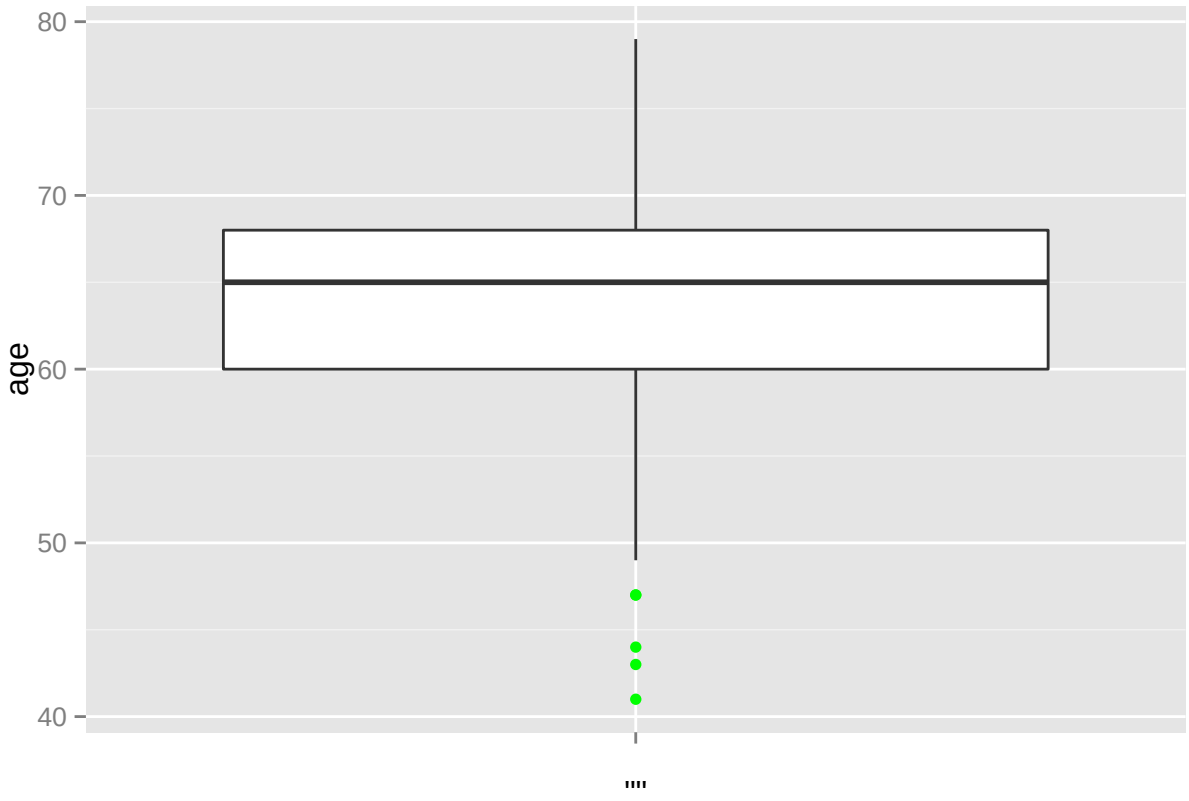
- vol: volume do câncer
- weight: peso do paciente
- age: idade do paciente
- bph: hiperplasia prostática benigna
- svi: invasão das vesículas seminais
- cp: penetração capsular
- gleason: escore Gleason
- pgg45: percentagem escore Gleason 4 ou 5
- psa: antígeno específico da próstata.

A ideia é, através de Regreção linear, criar um modelor apra prever os niveis de PSA no sangue do paciente.

Conhecendo Melhor os Dados

A idade dos pacientes avaliados é entre 41 e 79. A idade média é de 63.8659794. Como se pode observar no box plot abaixo, a maior parte deles tem idade entre 60 e 70 anos.

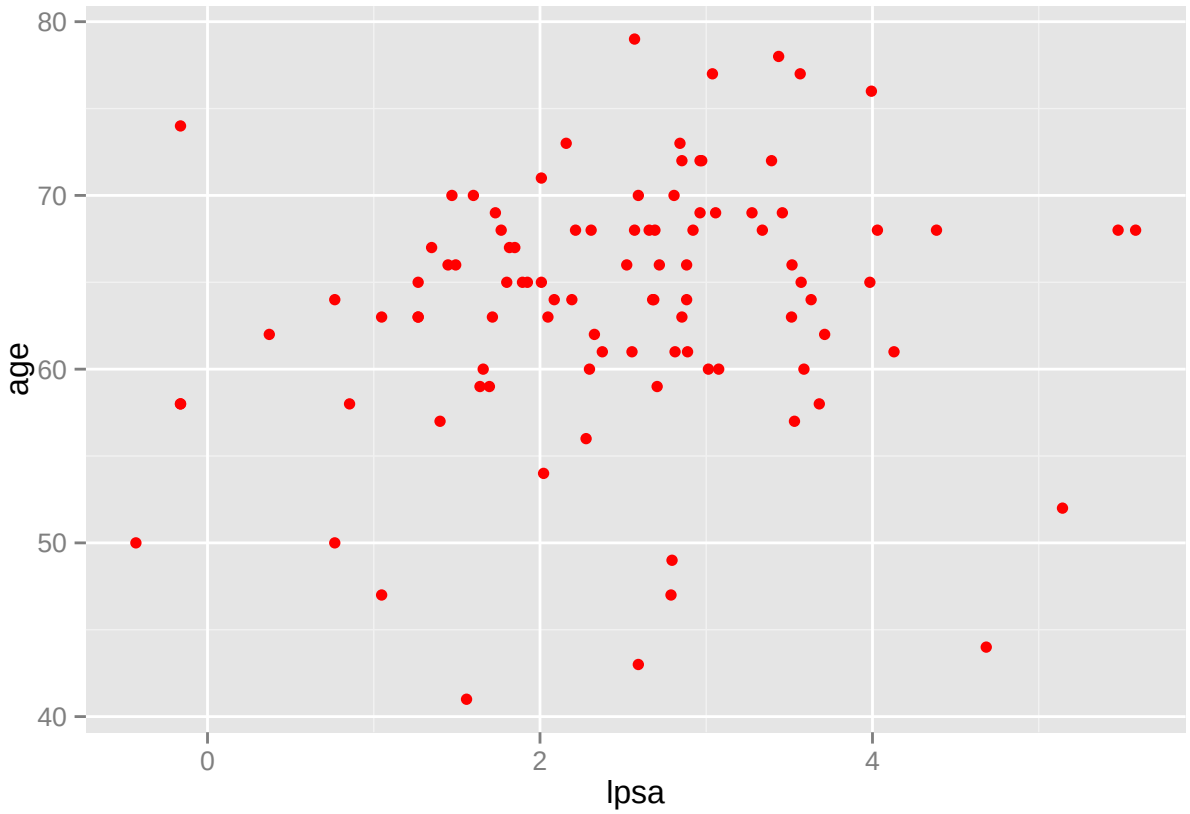
```
boxplot_age <- ggplot(prostate.data, aes(y=age, x = ""))
boxplot_age+geom_boxplot(outlier.colour = 'green')
```



Criando um primeiro Modelo

Muito se fala que homens a partir de uma certa idade devem começar a procurar um médico para prevenir o cancer de prostata. Um modelo inicial que pode ser feito é olhar a relação entre a idade e o nivel do PSA.

```
plt_agexlpsa <- ggplot(prostate.data, aes(lpsa, age))
plt_agexlpsa+geom_point(colour = "red", size = 2)
```



Antes de criar o modelo, vamos separar os dados em treino e teste. Será removido a variável X pois ela indica somente o número da linha da entrada. Após a separação será removido a variável train pois ela só serve para separar os dados.

```
prostate.data <- select(prostate.data, -X)

dados_treino <- prostate.data[prostate.data$train==TRUE,]
dados_treino$train <- NULL

dados_teste <- prostate.data[prostate.data$train==FALSE,]
dados_teste$train<- NULL
```

Vamos construir um modelo utilizando apenas a idade e lpsa (log do PSA).

```
reg_linear = lm(lpsa ~ ., select(dados_treino, age,lpsa))
summary(reg_linear)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = select(dados_treino, age, lpsa))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.95400 -0.67394  0.08716  0.70466  2.99243
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.07945    1.26726   0.063   0.9502
## age          0.03665    0.01944   1.885   0.0639 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.185 on 65 degrees of freedom
## Multiple R-squared:  0.05182,    Adjusted R-squared:  0.03723
## F-statistic: 3.552 on 1 and 65 DF,  p-value: 0.06393
```

O resultado desse modelo mostra que a variavel idade não ajuda a mensurar o PSA. Além de possuir um p-valor muito alto, o q significa que para aceitar a hipotese desse ser um bom modelo, é necessário um nivel de confiança muito baixo (nivel de confiança tem q ser menor que 99.93607%), isto indica que nosso modelo gera muitos erros.

Melhorando o Modelo

Vimos que a idade por si só não é capaz de prever o nível de PSA. Vamos agorar verificar como fica um modelo usando todas as 8 variáveis dos dados de treino

```
reg_multipla = lm(lpsa ~ ., dados_treino)
summary(reg_multipla)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = dados_treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64870 -0.34147 -0.05424  0.44941  1.48675
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.429170   1.553588   0.276  0.78334
## lcavol       0.576543   0.107438   5.366 1.47e-06 ***
## lweight      0.614020   0.223216   2.751  0.00792 **
## age         -0.019001   0.013612  -1.396  0.16806
## lbph         0.144848   0.070457   2.056  0.04431 *
## svi          0.737209   0.298555   2.469  0.01651 *
## lcp         -0.206324   0.110516  -1.867  0.06697 .
## gleason     -0.029503   0.201136  -0.147  0.88389
## pgg45        0.009465   0.005447   1.738  0.08755 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7123 on 58 degrees of freedom
## Multiple R-squared:  0.6944, Adjusted R-squared:  0.6522
## F-statistic: 16.47 on 8 and 58 DF,  p-value: 2.042e-12
```

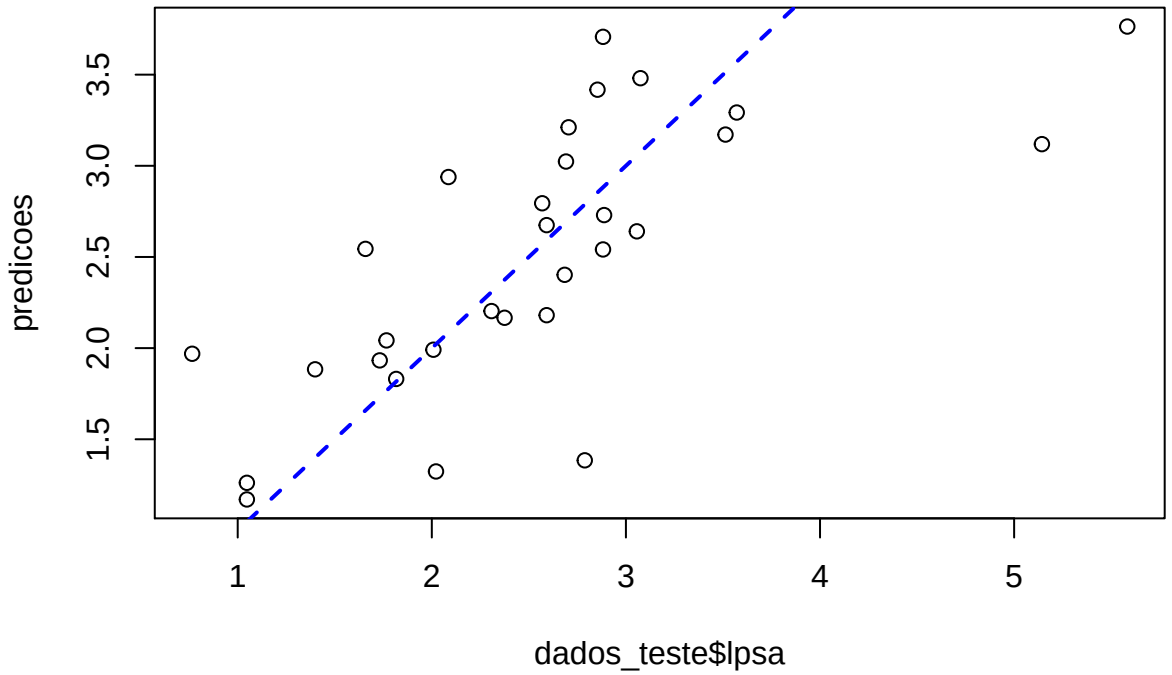
Ao contrário do modelo anterior esse modelo possui um p-valor pequeno. O que não nós permite rejeita-lo, com um bom nivel de significância, comoo um modelo para prever o PSA. O que iremos testar daqui para a frente é se ele é realmente um bom modelo, isto é, se os resultados das predições feitas por ele são bons, tem um erro baixo.

Começaremos inicialmente calculando as predições. Isto será feito usando os dados de teste. Também iremos calcular os residuos. Estes representam quanto o modelo errou, é a diferença entre o valor real e o valor predito pelo modelo.

```
predicoes = predict.lm(reg_multipla, dados_teste)
residuos = dados_teste$lpsa - predicoes
```

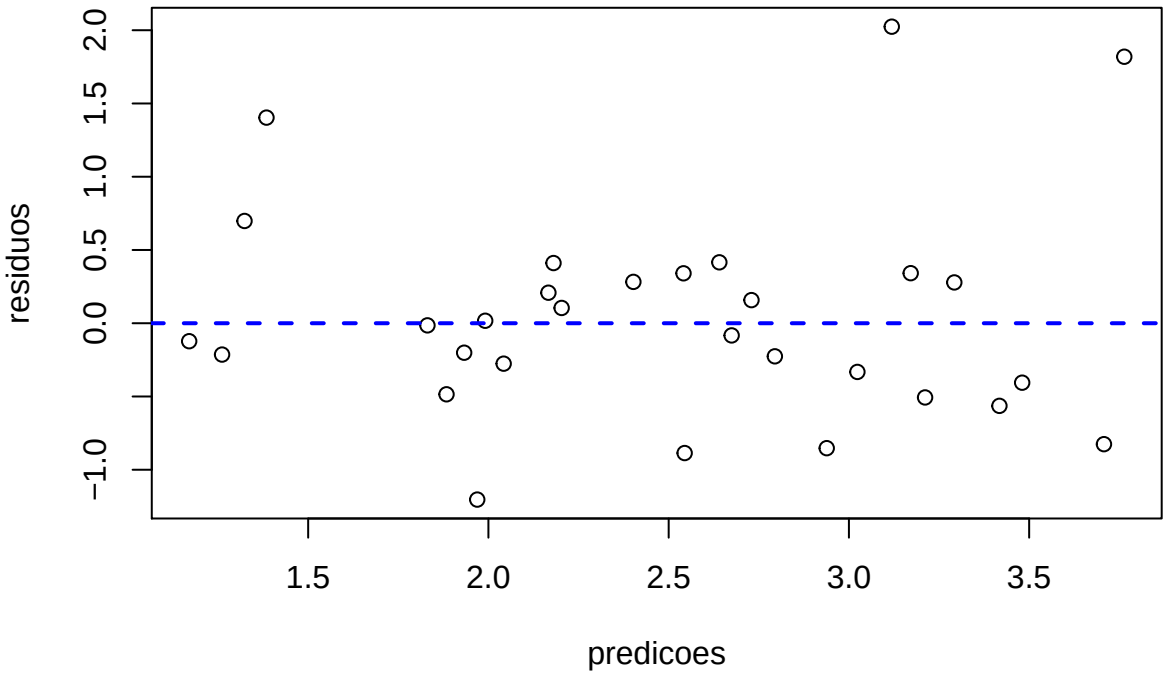
Abaixo podemos ver os dados reais versos os preditos

```
axisRange = extendrange(c(dados_teste$lpsa,predicoes))
plot(dados_teste$lpsa,predicoes)
abline(0,1,col="blue",lty=2,lwd=2)
```



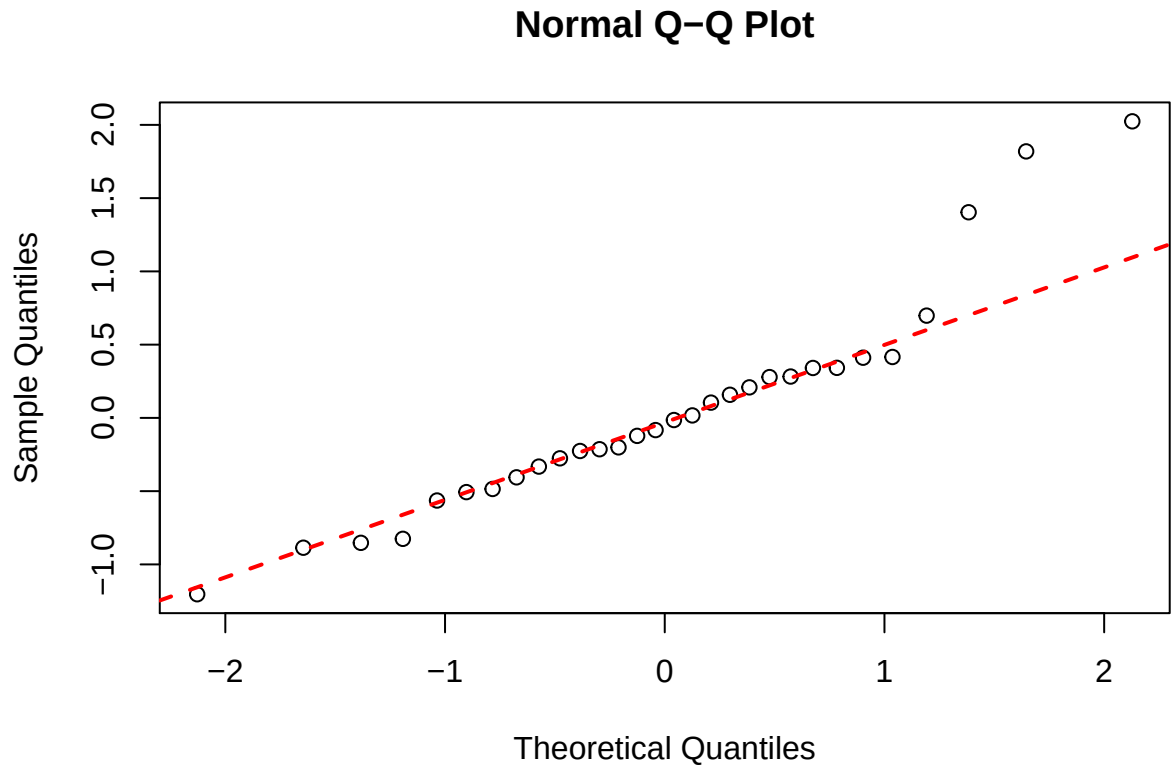
Com excessão de alguns outliers, os valores preditos, se aproximam dos valores reais. Outra forma de ver a relação acertos x erros é olhando o valor predito versus os residuos.

```
plot(predicoes,residuos)
abline(h=0,col="blue",lty=2,lwd=2)
```



Um forma de testar se temos um bom modelo, é olhando se os residuos seguem um distribuição normal com média 0. Uma forma, visual, de verificar isso é usando o gráfico do qq-plot. quanto mais próximos da linha da normal estiverem os dados, mais eles se apróximam de uma distribuição normal. Isso é o que acontece com a distribuição dos residuos, pode ser visto a baixo. Sendo assim temos um forte indicio de que temos um bom modelo.

```
qqnorm(residuos)
qqline(residuos, col = 2,lwd=2,lty=2)
```



Uma outra forma de vermos se temos um bom modelo é olhar o RMSE, quanto mais próximos de zero, melhor é o modelo analisado. Nosso modelo possui um RMSE de 0.7219931, o que é relativamente baixo, sendo assim podemos dizer que sim temos um modelo para prever o PSA.

Conclusão

Podemos dizer que encontramos um modelo que é capaz de prever o PSA apresentando um baixo nivel de erros. Não podemos dizer que este é o modelo perfeito para predizer, mas temos um modelo que causa baixos residuos.