21 DE DEZEMBRO DE 2018

Universidade do Minho
Escola de Engenharia

APRENDIZAGEM AUTOMÁTICA I
TRABALHO PRÁTICO

RELATÓRIO FINAL

Carlos José Lima Gonçalves, a77278
José Pedro dos Santos Ferreira, a78452
Ricardo Jorge Marques Peixoto, a78587

Conteúdo

Lista de Figuras

Lista de Tabelas

Resumo

Este relatório detalha o trabalho prático realizado na Unidade Curricular Aprendizagem Automática 1. Inicialmente é apresentado o conjunto de dados escolhido para realizar o problema e as perguntas às quais queremos dar resposta. Segue-se uma análise desses dados, sendo a mesma dividida em partes distintas. Após a realização da análise dos dados é explicada a forma como selecionamos o modelo, bem como os métodos utilizados, sendo posteriormente apresentado o modelo escolhido. Depois de termos escolhido o modelo respondemos às perguntas previamente colocadas e realizamos uma conclusão com uma análise ao trabalho realizado.

1 Introdução

De maneira a consolidar os conhecimentos obtidos na unidade curricular de AA1 foi-nos proposta a realização de um pequeno projeto que consistia em selecionar um *dataset* sobre um tema escolhido por nós. Posteriormente fizemos uma análise exploratória dos dados de maneira a selecionar apenas os mais significativos de maneira a trabalhar o *dataset* para retirarmos algumas conclusões sobre o mesmo.

2 Dataset escolhido

Neste projeto o grupo decidiu analisar o dataset Diabetes da biblioteca faraway. Este dataset incide sobre uma população afro-americana, nas cidades de Buckingham e Louisa, e tem como objetivo estudar a ocorrência de obesidade, diabetes e outros fatores cardiovasculares de risco. Para tal, o dataset apresenta 19 variáveis e 403 observações. As variáveis usadas neste dataset são as seguintes:

- id: ID da pessoa (não é considerado)
- chol: colesterol total
- stab.glu: glucose estabilizada
- hdl: Lipoproteína de alta densidade
- ratio: rácio entre colesterol e hdl
- glyhb: hemoglobina glicada
- location: cidade (um fator com opções Buckingham e Louisa)
- age: idade em anos
- gender: género (fator com opções male (masculino) e female (feminino))
- height: altura em polegadas
- weight: peso em libras
- frame: relação entre a circunferência do pulso relativamente à altura (um fator com opções small (pequeno), medium (médio) e large (grande))
- bp.1s: 1º pressão arterial sistólica (quando o coração bombeia sangue)
- bp.1d: 1º pressão arterial diastólica (quando o coração não está a bombear o sangue)
- bp.2s: 2º pressão arterial sistólica
- bp.2d: 2º pressão arterial diastólica
- waist: perímetro da cintura em polegadas
- hip: perímetro da anca em polegadas
- time.ppn: tempo, em minutos, após a refeição

A variável resposta depende do nível de hemoglobina glicada dado que para valores superiores a 7 o teste de diabetes é geralmente considerado positivo. Ao observarmos estes dados vemos que se trata de um problema supervisionado, de classificação. Podemos, portanto, realizar este projeto utilizando técnicas de classificação nas quais obtemos a probabilidade de ter diabetes. No entanto, e visto que conhecemos o método de classificação de ter ou não diabetes, podemos também usar um método de regressão para prever o valor da hemoglobina e depois classificar consoante o valor obtido.

3 Análise exploratória

Antes de passar para a seleção do melhor modelo para o objetivo traçado, prever se um indivíduo tem ou não diabetes, foi importante analisar os diferentes dados presentes no *dataset* escolhido. Ao analisar os dados verificamos que

existiam alguns preditores que nos eliminavam bastantes linhas, reduzindo o dataset para perto dos 35%, pelo que decidimos não considerar os mesmos. Como podemos verificar na imagem acima esses dois eram:

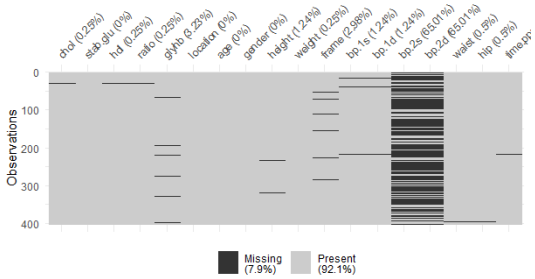


Figura 1: Valores em falta no *dataset*

- bp.2s
- bp.2d

Para além de termos verificado os valores em falta, analisamos também as diferentes correlações entre as variáveis, para não incluirmos no mesmo modelo 2 preditores que estão altamente correlacionados. Quando uma variável está altamente correlacionada com outra é porque elas se explicam mutuamente e na maior parte dos casos não se obtêm ganhos significativos ao colocar ambas ao invés de apenas uma, aumentando apenas a complexidade do modelo. Esta análise serviu também para perceber quais preditores estavam mais correlacionados com a resposta **glyhb**.

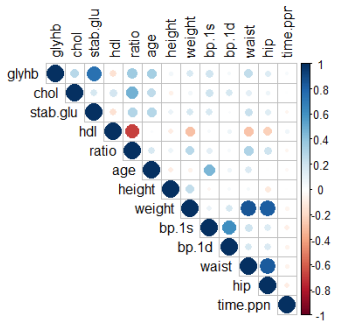


Figura 2: Correlações entre as variáveis

Analisando as diferentes correlações identificamos as seguintes como importantes e significativas entre os preditores:

- weight-waist = 85
- weight-hip = 83
- waist-hip = 83

Para além destas identifcamos as correlações importantes com a variável resposta, sendo que apenas identificamos a seguinte:

- glyhb-stab.glu = 74% (não sendo tão importante como as outras, foi a mais importante e única relevante com a variável resposta)

Visto que a base de dados escolhida continha bastantes preditores, foi necessário perceber, logo de início, quais os realmente importantes para a variável resposta. Para perceber a importância de cada preditor na variável resposta realizamos uma análise de influência de cada preditor tanto na resposta qualitativa como na quantitativa. Esta análise

à inflência está detalhada posteriormente sendo que nos permitiu eliminar um número considerável de preditores e perceber os que são realmente importantes para o modelo.

Como alguns dos métodos que pretendíamos explorar para selecionar o melhor modelo, *QDA* e *LDA*, se baseiam nos pressupostos de que os preditores seguem uma distribuição normal decidimos realizar uma análise à normalidade das variáveis. Esta análise serviu para identificar os preditores que podiam ser utilizados nos métodos referidos e está detalhada posteriormente.

Para além da nossa análise tivemos também em conta a análise do comando *regsubsets*, utilizando diferentes métodos. No entanto não vamos detalhar essa análise neste relatório, falando apenas posteriormente quais as variáveis utilizadas. Em anexo segue o código, onde pode ser possível visualizar os comandos utilizados.

3.1 Análise de influência

De maneira a perceber a importância individual de cada variável fizemos uma análise de influencia em relação à variável de resposta binária e variável de resposta quantitativa.

Para isso utilizamos diferentes tipos de gráficos, sendo que o mais significativo para perceber a importância da variável em relação à variável binária foi a caixa de bigodes, e um gráfico de pontos para a variável continua.

3.1.1 Variável binária

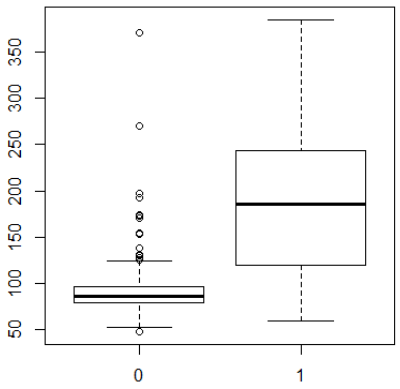


Figura 3: Stab.glu

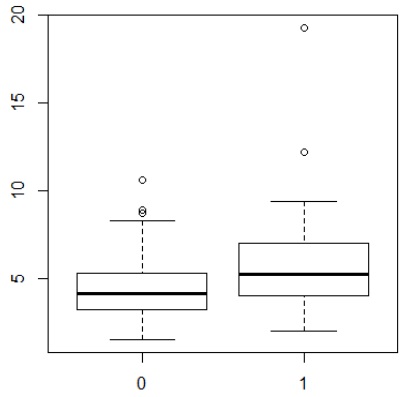


Figura 4: Ratio

De seguida mostrámos os gráficos das quatro variáveis que revelaram mais influência na variável resposta binária, sendo que podemos observar que a stab.glu é claramente aquela que se destaca das demais a nível de importância.

3.1.2 Variável contínua

Na imagem em cima apresentada podemos observar as variáveis que se mostraram mais significativas em relação à variável de resposta continua. Uma vez mais a stab.glu é a que apresenta uma relação mais vincada de sentido linear e crescente. As restantes também aparentam um comportamento similar, apesar de não ser tão percetível através da análise dos gráficos.

3.1.3 Rankings

Após a análise referida anteriormente decidimos criar um *ranking* para cada variável de resposta para que fosse claro quais as variáveis mais significativas. Contrariamente ao que suspeitávamos inicialmente, apesar da variável binária ter derivado da variável continua, os *rankings* apresentam algumas diferenças.

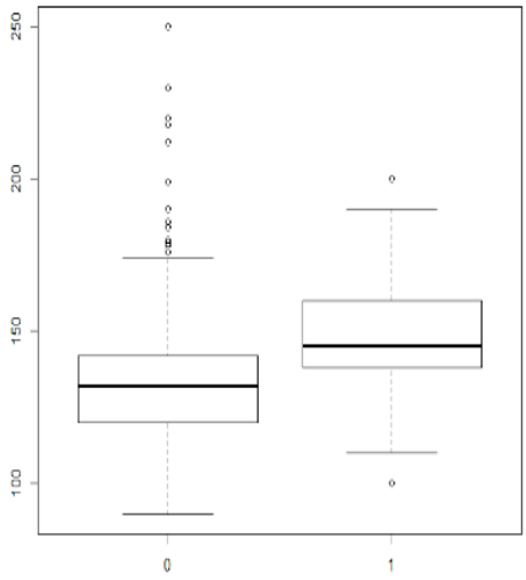


Figura 5: Bp.1s

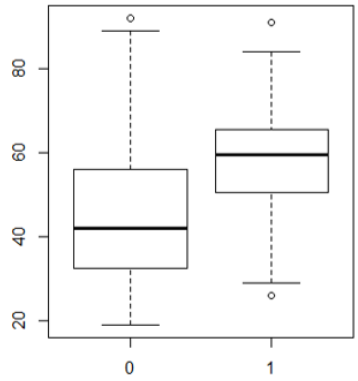


Figura 6: Age

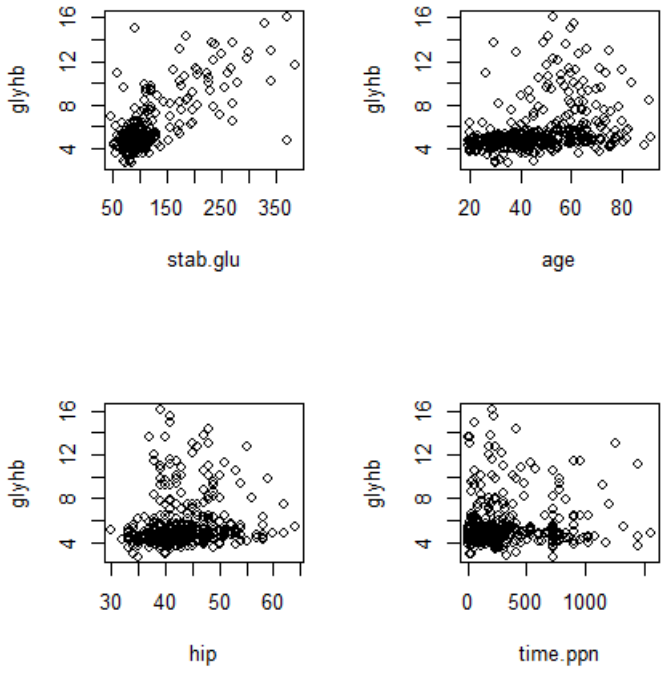


Figura 7: Gráfico 4 vars mais importantes

Variável Binária:

- **1º** Stab.glu
- **2º** Age
- **3º** Bp.1s
- **4º** Racio
- **5º** Waist

- **6º** Weight
- **7º** Hip
- **8º** Hdl
- **9º** Chol
- **10º** Bp.1d

Variável Continua:

- **1º** Stab.hlu
- **2º** Age
- **3º** Time.ppn
- **4º** Hip
- **5º** Bp.1s
- **6º** Waist
- **7º** Bp.1d
- **8º** Chol
- **9º** Height
- **10º** Weight

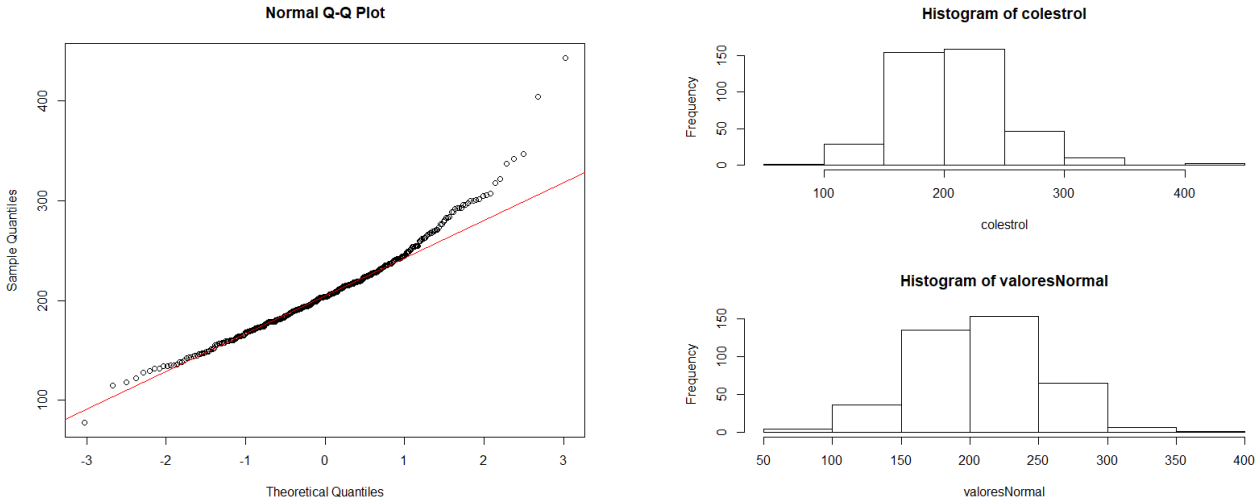
Apesar de termos ordenado todas as variáveis (17) decidimos mostrar apenas as primeiras dez, visto que garantidamente os nossos modelos não precisariam das variáveis menos importantes.

3.2 Análise à normalidade

Por forma a excluir alguns métodos que pressupõem certas distribuições da variável resposta e/ou preditores, procedemos a um teste à normalidade de cada uma das variáveis quantitativas do nosso modelo. Para isso, comparamos os histogramas de cada variável com o histograma dos valores gerados segundo uma distribuição normal seguindo a média e desvio padrão dessa variável. Além disso, comparamos ainda com os gráficos de teste da normalidade. Realizamos também *shapiro tests* mas estes não produziram resultados interessantes. De todos os testes efetuados, decidimos dar mais importância à distribuição apresentada nos histogramas dado que os resultados obtidos são mais explícitos.

Neste capítulo apresentamos apenas a análise efetuada a algumas das variáveis (*chol*, *stab.glu* e *ratio*) e a metodologia seguida nas mesmas, uma vez que o processo de análise às restantes variáveis segue a mesma lógica e as respetivas conclusões inferidas já são cobertas pelas três variáveis apresentadas. Deste modo, seguem em anexo as imagens dos histogramas e dos gráficos de teste da normalidade das variáveis que não são discutidas neste capítulo.

Colesterol – chol



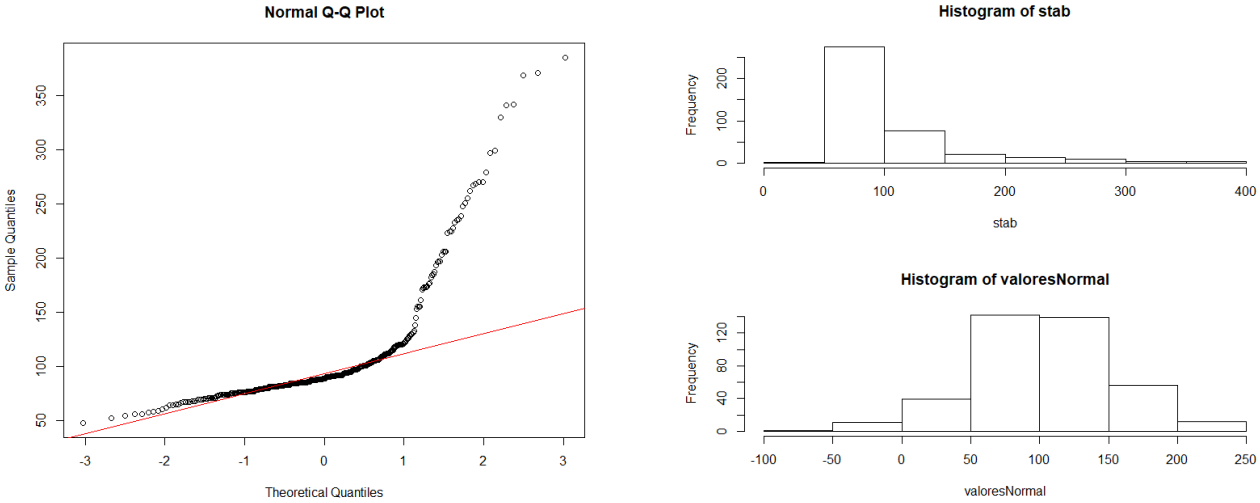
(a) Q-Q Plot

(b) Histogramas

Figura 8: Chol

Pela análise do gráfico do teste da normal podemos verificar que esta variável é relativamente normal, visto que a distribuição dos resíduos se aproxima bastante da linha vermelha. Pela análise do histograma da variável e do histograma da distribuição normal simulado com os valores da variável podemos verificar que são relativamente simétricos. Isto indica que a variável aparenta possuir o comportamento de uma variável normal.

Glucose estabilizada – stab.glu



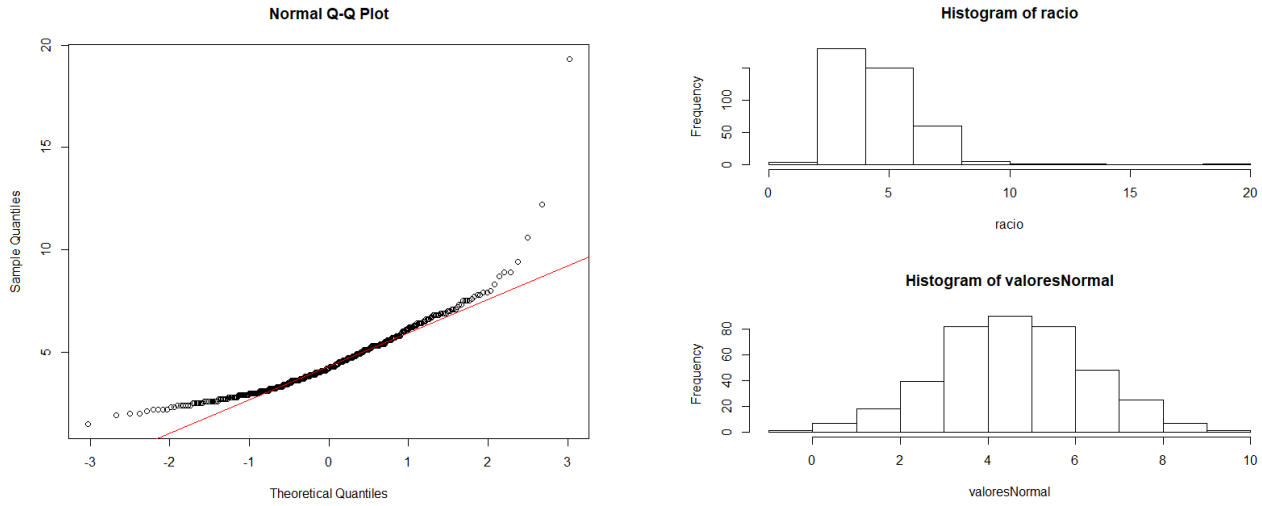
(a) Q-Q Plot

(b) Histogramas

Figura 9: Stab.glu

Pela análise do gráfico do teste da normal podemos verificar que esta variável claramente não apresenta o comportamento de uma variável normal. Pela análise do histograma da variável e do histograma da distribuição normal simulado com os valores da variável podemos verificar que o histograma da variável é bastante assimétrico e em nada se assemelha ao histograma simulado, confirmando que esta variável não segue uma distribuição normal.

Rácio Colesterol/HDL – ratio



(a) Q-Q Plot

(b) Histogramas

Figura 10: ratio

Pela análise do gráfico do teste da normal podemos verificar que esta variável é relativamente normal, visto que em certas partes a sua distribuição aproxima-se bastante da linha vermelha, apesar de noutras se afastar bastante. Pela análise do histograma da variável e do histograma da distribuição normal simulado com os valores da variável podemos verificar que o histograma da variável é assimétrico e não se assemelha muito ao histograma simulado. Deste modo a variável não aparenta seguir uma distribuição normal

Da análise efetuada concluímos que apenas as seguintes variáveis apresentam uma distribuição semelhante a uma distribuição normal:

- chol
- hdl
- age
- height
- weight
- bp.1d
- waist
- bp.1s

Importante referir que a variável *stab.glu* não aparenta seguir uma distribuição normal. Na análise da influência é possível observar que esta variável é considerada a mais influente, tanto ao nível da variável resposta binária como ao nível da variável resposta *glyhb*. Assim sendo, esta análise mostra que métodos que exijam que as variáveis sigam uma distribuição normal não vão produzir bons resultados uma vez que os seus modelos não vão poder incluir a variável mais influente, *stab.glu*.

4 Seleção do modelo

Depois de termos realizado a análise exploratória, onde exploramos os dados e as variáveis do *dataset*, estávamos prontos para a próxima fase: a seleção do melhor modelo para a previsão dos diabetes. Para a seleção deste modelo decidimos dividir o trabalho em 2 fases distintas:

1. Dividimos o trabalho em 3 sub-partes:

   - Regressão Linear
   - Regressão Logística
   - KNN, QDA e LDA

   Sendo que a ideia passou por selecionar 1 (ou 2) melhores modelos de cada uma das sub-partes, para posteriormente serem comparados e selcionar o melhor modelo global.

2. Seleção do melhor modelo global, consoante os melhores modelos selecionados nas diferentes sub-partes

Relativamente às comparações entre os modelos decidimos não utilizar apenas o acerto como fator de decisão, mas também ter em conta a complexidade do modelo (quanto mais baixa, melhor), assim como a sua interpretabilidade (quanto mais alta, melhor). O acerto não é apenas um acerto normal, mas sim um acerto ponderado, sendo que será explicado com mais detalhe mais tarde neste relatório.

4.1 Diferentes abordagens

Na seleção do melhor modelo para cada um dos métodos especificados anteriormente tivemos essencialmente 3 fases distintas.

Numa primeira fase em todos os métodos os dados de treino e dados de teste eram os mesmos e correspondiam ao *dataset* completo. No entanto esta abordagem beneficia modelos com *overfitting* nos quais o modelo se ajusta demasiado aos dados. Isto seria preferível se soubéssemos que os dados não contêm ruídos e tivéssemos poucos dados e muitos preditores. Contudo, nada sabemos sobre os ruídos presentes nos dados e pela análise exploratória conseguimos verificar que os preditores realmente importantes seriam um número reduzido, pelo que ficamos com poucos preditores e um número considerável de dados (relativamente aos preditores), pelo que não consideramos esta a melhor abordagem e decidimos alterar.

Numa segunda fase passamos a diferenciar os dados de treino e teste, para ultrapassar o problema referido anteriormente, sendo esta diferenciação realizada de uma forma aleatória. No entanto, esta aleatoriedade na escolha dos dados de treino não é positiva visto que é provável que o acerto contenha uma variância elevada, devido a que ao treinar com um subconjunto os dados podem ter um acerto elevado e o melhor modelo ser um e ao treinar com outro as conclusões serem completamente diferentes. Como esta abordagem é muito instável decidimos voltar a alterar a abordagem escolhida.

Na terceira e última fase passamos a utilizar *k-folder cross validation*. Nesta abordagem os dados de teste são divididos em pedaços de k elementos e os de treino os restantes N-k, sendo que (quase) todo o dataset serve como dados de teste visto que é feita uma "medição"do acerto para os N/k diferentes conjuntos dos dados de teste. Para calcular o acerto utiliza-se a média de todos os acertos calculados, sendo por isso um método bastante mais estável. Por estes fatores consideramos esta abordagem como a melhor e utilizamos a mesma para a seleção dos melhores modelos para cada sub-parte referida anteriormente. De referir que criamos umas funções que realizam validação cruzada para o **glm, qda e lda** com k=10, porque estudos comprovam como sendo um k "bastante bom"e utilizamos a função fornecida pelo **R** para o **KNN**, com k=1 por nos poupar trabalho e parecer também uma boa solução. Para a regressão linear utilizamos uma função fornecida pelo **R** com k=10.

4.1.1 Regressão linear

Para a regressão linear, começamos por definir dois subconjuntos de variáveis que seguiam uma ordem de influência. Um deles foi obtido através da função "regsubsets" e o outro foi criado pelo grupo. De seguida analisamos as variáveis que iriam ser usados nos modelos. Como na fase da análise exploratória vimos que a partir do quinto preditor a

influência sobre a variável resposta já não é significativa, optamos por definir apenas os 5 preditores mais influentes em cada um destes subconjuntos. O processo para seleção do melhor modelo pode ser dividido em três fases:

- Análise dos preditores
- Construção dos modelos
- Seleção do melhor modelo

Análise dos preditores

Inicialmente começamos por realizar uma análise das distribuições dos resíduos dos vários preditores para entender de que forma estes preditores afetavam a variável resposta. Para tal recorremos aos gráficos *Residuals vs Fitted*, *Normal Q-Q*, *Scale-Location* e *Residuals vs Leverage*. Esta abordagem é importante para nos ajudar a compreender de que forma as variáveis vão influenciar o modelo antes de serem adicionados ao mesmo e consequentemente ajudar a compreender a causa das mudanças que modelo vai sofrer após estas variáveis serem adicionados.

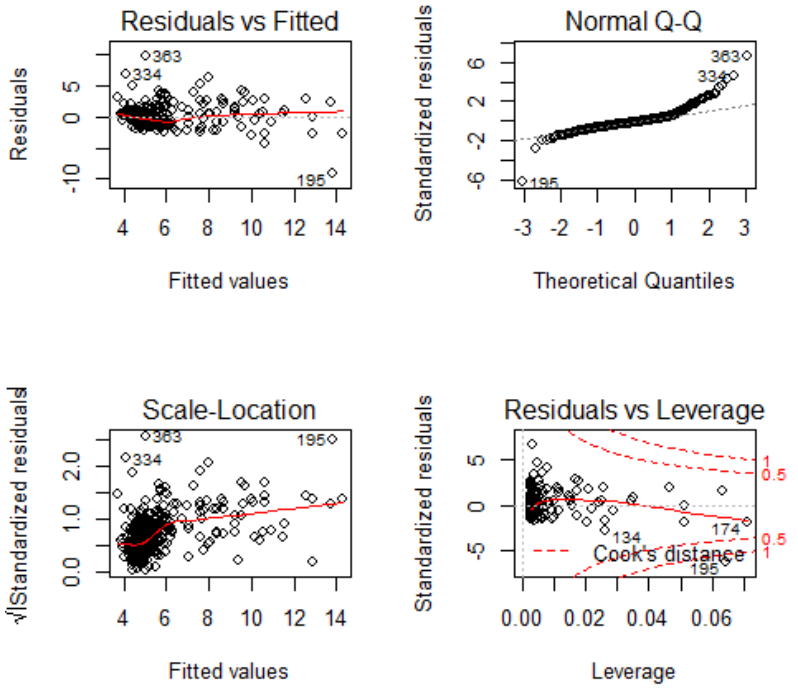


Figura 11: Resíduos

Através do gráfico *Residuals vs Fitted* verificamos se existem ou não relações não lineares entre um dado preditor e a variável resposta observando a distribuição dos resíduos. Se estes estiverem igualmente espalhados ao longo de uma linha "reta"(neste caso a linha a vermelho) podemos inferir que em princípio a relação entre o preditor e a variável resposta é linear.

Já com o gráfico *Normal Q-Q* podemos verificar se o preditor segue ou não uma distribuição normal com a variável resposta. Se os resíduos seguirem uma linha "reta"então o preditor segue distribuição normal. No caso da regressão linear este gráfico não se mostra muito interessante pois esta regressão irá também retratar relações não lineares nos seus modelos.

O gráfico *Scale-Location* informa-nos se os resíduos estão bem distribuídos na gama de valores do preditor. Quanto mais horizontal se apresentar a linha vermelha, melhor distribuídos se encontram os resíduos.

Por último, o gráfico *Residuals vs Leverage* é útil para encontrar casos que alteram drasticamente o resultado caso sejam removidos do *dataset* (*outliers*). Na imagem acima podemos identificar o caso 195 como sendo um *outlier*. Com esta análise podemos remover casos não significativos que estejam a afetar drasticamente os nossos modelos.

Após esta análise, verificamos que todas as variáveis apresentam uma relação linear com a variável resposta e algumas delas não apresentam uma distribuição muito semelhante à distribuição normal. Vimos também que algumas variáveis possuem uma má distribuição de resíduos ao longo da sua gama de valores. Por fim, foram detetados poucos *outliers* e num número muito reduzido de variáveis. As imagens referentes a cada uma das variáveis desta análise encontram-se em anexo.

Apesar de todas as variáveis apresentarem uma relação linear com a variável resposta, decidimos testar também modelos com relações quadráticas e cúbicas.

Construção de modelos

Na criação dos modelos, decidimos começar pelo modelo mais simples usando a variável mais influente *stab.glu* e adicionar sequencialmente complexidade ao modelo até não ser mais possível obter um ganho significativo que compense o aumento da complexidade.

Numa fase inicial começamos por comparar modelos mais simples com modelos mais complexos tendo por base o *Adjusted R-squared*, ou seja, o modelo que melhor representasse os dados. Esta abordagem rapidamente foi abandonada pois conduz a situações de *overfitting* onde o modelo representa muito bem os dados que foram usados para o treinar mas face a novos dados a qualidade da sua resposta decai consideravelmente.

Numa segunda abordagem, recorremos adicionalmente ao *partial F-test* e à **validação cruzada** para comparar um modelo mais simples com um modelo mais complexo. O *partial F-test* apenas indica se aceitamos ou rejeitamos a hipótese nula de que o modelo complexo não é significativamente melhor que o modelo simples, ou seja, não podemos afirmar que o modelo complexo é significativamente melhor tendo por base apenas este teste.

```
> anova(lm.fit1, lm.fit2)
Analysis of Variance Table

Model 1: glyhb ~ stab.glu
Model 2: glyhb ~ poly(stab.glu, 2, raw = TRUE)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    388 858.15
2    387 841.15  1        17 7.8216 0.00542 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 12: Partial F-test

No caso apresentado na figura acima rejeitamos a hipótese nula de que o modelo 2 não é significativamente melhor que o modelo 1.

Para vermos se efetivamente o mais complexo é melhor que o modelo menos complexo recorremos à **validação cruzada**, mais propriamente ao método *10K-CV*. Este método de validação cruzada permite evitar questões de *overfitting* pois vai dividir a base de dados em 10 partes e iterativamente treinar o modelo com 9 das partes e verificar o desempenho do mesmo face à parte que deixou de fora. Este processo iterativo termina quando todas as partes ficarem de fora (10 iterações) e a avaliação do modelo tem por base estas 10 iterações. Nas imagens seguintes podemos ver o código em R do processo de treino e comparação dos modelos usando o método *10K-CV*:

```
## validação cruzada
train.control <- trainControl(method = "cv", number = 10)
set.seed(1)
model1 <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
set.seed(1)
model2 <- train(glyhb~poly(stab.glu, 2, raw=TRUE), data = treino, method = "lm", trControl = train.control)
models <- resamples(list("stab.glu(grau1)"=model1,
                         "stab.glu(grau2)"=model2))
summary(models)
```

Figura 13: Criação dos modelos

```
> summary(models)

Call:
summary.resamples(object = models)

Models: stab.glu(grau1), stab.glu(grau2)
Number of resamples: 10

MAE
                    Min.   1st Qu.    Median      Mean  3rd Qu.     Max. NA's
stab.glu(grau1) 0.7083120 0.8638963 0.9214334 0.9494561 1.038206 1.178243    0
stab.glu(grau2) 0.7294652 0.8604003 0.9414666 0.9656313 1.074323 1.232015    0

RMSE
                    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
stab.glu(grau1) 0.8924899 1.237089 1.317318 1.454001 1.561458 2.179360    0
stab.glu(grau2) 0.9153366 1.249119 1.325275 1.456322 1.563056 2.129171    0

Rsquared
                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
stab.glu(grau1) 0.1945056 0.3721575 0.6562839 0.5689260 0.7314516 0.8267779    0
stab.glu(grau2) 0.1875656 0.3890249 0.6320048 0.5649829 0.7333875 0.8242778    0
```

Figura 14: Comparação dos modelos

Na última imagem podemos ver várias medidas do *Mean Absolute Error*, do *Root Mean Squared Error* e do *R-squared* calculadas através das 10 iterações da *10K-CV* para os dois modelos. Neste caso o melhor modelo é então o modelo do polinómio de grau 1 pois é menos complexo, apresenta em média menor MAE, menor RMSE com um máximo pouco superior ao modelo mais complexo e maior R-squared, ou seja, o modelo mais simples responde melhor a novos dados e apresenta erros menores.

Seleção do melhor modelo

Apresentamos de seguida os modelos da regressão linear que apresentaram um melhor desempenho a representar o *dataset* e a sua respetiva avaliação através do método *10K-CV*:

Fórmula	MAE	RMSE	R^2	Adj. R^2
Stab.glu	0.9495	1.4540	0.5689	0.5678
Stab.glu^3	0.9389	1.4878	0.5632	0.5598
Stab.glu + age	0.9390	1.4257	0.5887	0.5865
stab.glu + age + ratio	0.9332	1.4179	0.5978	0.5947

Observando esta tabela podemos verificar que o quarto modelo é o que apresenta menor *Mean Absolute Error* mas o *MAE* do segundo e terceiro modelos é bastante próximo. O *Root Mean Squared Error* apresenta-se mais baixo também no quarto modelo mas o terceiro modelo apresenta valores muito próximos. Já os dois primeiros modelos apresentam um *RMSE* muito superior. Tendo em conta os erros obtidos, é normal o quarto modelo apresentar erros mais baixos que o terceiro modelo por ser um modelo mais complexo.

Por fim, podemos ver que o *Adjusted R-squared* do terceiro modelo é bastante superior ao dos modelos anteriores e pouco inferior ao do quarto modelo.

Tendo em conta estas informações, podemos dizer que os dois últimos modelos são significativamente melhores que os dois primeiros. Como o quarto modelo não é significativamente melhor que o terceiro modelo tanto a nível dos erros apresentados como a nível da explicação dos dados (*Adjusted R-squared*), decidimos que não era vantajoso aumentar a complexidade do modelo para obter um ganho tão marginal. Deste modo, selecionamos o modelo *Stab.glu + age* como sendo o melhor modelo para a regressão linear de entre todos os modelos que testamos.

4.1.2 Regressão logística

Para o modelo de regressão logística utilizamos como variável de resposta a variável binária criada, como referido anteriormente. Para a seleção dos preditores consideramos as respostas obtidas através da análise de influencia, e utilizamos também as variáveis indicadas pelo RegSubSets. De seguida combinamos as variáveis através de diferentes métodos, interações e polinómios e selecionamos os que apresentavam um melhor acerto ponderado, que será explicado

de seguida.

Acerto ponderado

Para avaliarmos o desempenho dos modelos escolhidos começamos por considerar apenas a taxa de acerto simples. No entanto chegamos à conclusão que seria sensato considerar também casos em que o modelo acertou nas previsões positivas e negativas (ter ou não ter diabetes). Passamos então a utilizar um acerto ponderado, que não varia de 0 a 1, mas de 0 a 4, e que pode ser calculado da seguinte maneira:

$$acertoPonderado = acertoTotal * 2 + acertoNegativos + acertoPositivos$$

Para cada modelo escolhido decidimos criar uma função que variava o *threshold* de 0 a 1 em intervalos definidos por nós de maneira a encontrar o valor que maximizava o acerto ponderado.

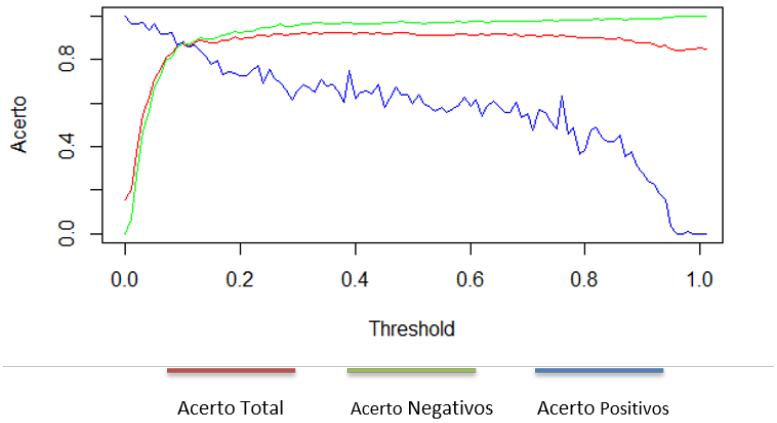


Figura 15: Gráfico threshold

Apesar de pelo gráfico poder parecer que o valor do *threshold* ideal se encontra entre o 0.1 e o 0.2, como utilizamos um acerto ponderado o valor que maximizou o acerto rondava os 0.39.

Melhores modelos
Apresentamos de seguida os modelos da regressão logística que apresentaram um melhor desempenho.

Fórmula	Thr	AcertoP
Stab.glu^2	0.39	3.5714
Stab.glu^2+ratio	0.39	3.5714
Stab.glu+ratio + stab.glu x ratio	0.39	3.5520

Podemos observar que o modelo vencedor é o mais simples, visto que apenas inclui a variável *stab.glu* elevada ao quadrado. Não foi surpreendente porque já tínhamos denotado esta importância na análise de influencia feita anteriormente. Em segundo lugar ficou o modelo que soma ao modelo inicial a variável *racio*, que apesar de ter o mesmo desempenho que o modelo anterior é mais complicado e por isso ficou em segundo. Por fim temos um modelo com uma interação entre a stab.glu e o ratio que também obteve uma performance bastante razoável. De salientar apenas que estes valores foram obtidos utilizando a metodologia de *cross validation.*

4.1.3 KNN

Para verificar qual o melhor modelo relativo ao método dos **K vizinhos mais próximos**, utilizamos como já foi referido *k-folder cross validation* com k=1, pois o **R** já nos fornecia a função. Para além disto, decidimos ter em conta tanto a nossa análise dos dados, como a do *regsubsets* pelo que dividimos o trabalho em 2 partes distintas:

- Nossa análise
- Análise do *regsubsets*

O método utilizado foi bastante simples, pois tentamos começar a usar apenas uma variável (a mais importante dada pela análise) e fomos acrescentando variáveis pelo seu grau de importância, até verificarmos que não valia a pena continuar. Para cada "modelo"(com as variáveis definidas) utilizamos uma estratégia parecida à do **glm**, no qual em vez de variarmos o *threshold* variamos o **k** (dos vizinhos mais próximos) de 1 até 20 e guardamos num *array* os seguintes valores:

- k
- Acerto
- Acerto nos positivos
- Acerto nos negativos

Para isto criamos uma função que consoante os dados de treino que lhe são passados (preditores e resposta), varia o **k** entre os valores passados como inicio e fim (usamos entre 1 e 20) e para cada k guarda no *array* os valores referidos acima, utilizando a validação cruzada para encontrar o valor das respostas.
Para a seleção do melhor modelo criamos uma função que nos dá o melhor **k** para cada um (com as diferentes variáveis), consoante o melhor acerto ponderado (explicado anteriormente). Criamos também uma função que para cada **k** de cada modelo dá um *array* com o valor do k, o "indíce"do modelo e o valor do acerto ponderado, depois ordena por ordem decrescente de acerto e retorna o respetivo *array.* Para perceber qual o melhor modelo basta analisar o *array* dado.
Como pode ser visto pelas tabelas acima, não prosseguimos tanto na nossa análise, como na do *regsubsets* visto que os valores do acerto ponderado não estavam a subir, ou então o ganho passou a ser reduzido. Por estas razões sentimos que não valia a pena trazer ainda mais complexidade para o modelo, pois o **knn** já é complexo por si só, e decidimos escolher o melhor consoante estes modelos criados.

Tabela 1: Melhores resultados para cada modelo, sendo nossa análise

Variáveis	K	Acerto Ponderado
Stab.glu	14	3.4825
Stab.glu+age	9	3.4707
Stab.glu+age+bp.1s	10	3.5320
Stab.glu+age+bp.1s+ratio	12	3.5316
Stab.glu+age+bp.1s+ratio+waist	10	3.5307

Tabela 2: Melhores resultados para cada modelo, sendo análise regsubsets

Variáveis	K	Acerto Ponderado
Stab.glu	14	3.4825
Stab.glu+chol	9	3.4902
Stab.glu+ratio+age	16	3.4766
Stab.glu+ratio+age+time.ppn	3	3.4286
(F)stab.glu+chol+age	8	3.4684
(F)stab.glu+chol+age+time.ppn	4	3.4014
(B)stab.glu+ratio	15	3.4820

	indice	k	valor
50	3	10	3.532048
52	3	12	3.532048
72	4	12	3.531597
90	5	10	3.530688
45	3	5	3.520823
51	3	11	3.510186
70	4	10	3.509722
71	4	11	3.509722
49	3	9	3.501915
69	4	9	3.501427
88	5	8	3.500445
89	5	9	3.500445
66	4	6	3.498418
53	3	13	3.496597
54	3	14	3.496597
73	4	13	3.496142
74	4	14	3.496142
63	4	3	3.495409
44	3	4	3.490689
43	3	3	3.487736
91	5	11	3.486884
92	5	12	3.486884
86	5	6	3.483763
14	1	14	3.482517
15	1	15	3.482517
16	1	16	3.482517

Figura 16: Melhores valores globais para a nossa análise

	indice	k	valor
29	2	9	3.490211
14	1	14	3.482517
15	1	15	3.482517
16	1	16	3.482517
18	1	18	3.482517
135	7	15	3.482030
56	3	16	3.476584
12	1	12	3.468881
30	2	10	3.468403
48	3	8	3.468403
49	3	9	3.468403
50	3	10	3.468403
51	3	11	3.468403
52	3	12	3.468403
53	3	13	3.468403
54	3	14	3.468403
59	3	19	3.468403
88	5	8	3.468403
23	2	3	3.465668
34	2	14	3.462956
13	1	13	3.460723
17	1	17	3.460723
19	1	19	3.460723
20	1	20	3.460723
27	2	7	3.460222
47	3	7	3.460222
55	3	15	3.460222

Figura 17: Melhores valores globais para a análise do regsubsets

(nas imagens o índice corresponde ao índice na tabela)
Podemos verificar nas imagens que os resultados obtidos pela nossa análise foi bastante superior aos obtidos pela análise do *regsubsets*. Por isso, selecionamos os 2 melhores modelos da nossa análise, ou seja:

- diabetesB stab.glu+age+bp.1s, k=10
- diabetesB stab.glu+age+bp.1s, k=12

Decidimos escolher estes dois pois sãos os melhores globais e dos melhores globais são os menos complexos.

4.1.4 QDA e LDA

Relativamente aos métodos **QDA** e **LDA** o nosso grupo desconsiderou-os logo à partida como sendo possíveis melhores métodos pelo facto de estes assentarem no pressuposto da normalidade dos preditores. Estes pressupostos levaram-nos a excluir desde logo os métodos visto que a variável que mais influencia a resposta, de longe, não segue uma distribuição normal, pelo que não a pudemos usar. No entanto, apenas como um caso de teste decidimos realizar alguns modelos para verificar os resultados.

Tabela 3: Resultados para o método QDA

Variáveis	Threshold	Acerto Ponderado
Age	0.22	2.8265
Age+bp.1s	0.24	2.8260
Age+bp.1s+waist	0.18	2.9115

Tabela 4: Resultados para o método LDA

Variáveis	Threshold	Acerto Ponderado
Age	0.18	2.8578
Age+bp.1s	0.26	2.8039
Age+bp.1s+waist	0.16	2.9364

Como podemos verificar pelas tabelas acima, os resultados obtidos foram muito baixos em comparação com os restantes, o que já esperávamos, pelo que não decidimos considerar estes métodos importantes. Não decidimos continuar a analisar estes métodos dado que pelas razões explicadas anteriormente os resultado não seriam significativamente melhores.

4.2 Modelo escolhido

Depois de termos verificado quais os melhores modelos para cada método, individualmente e com "auxílio"da validação cruzada, decidirmos comparar esses modelos para selecionar o melhor deles todos.

Decidimos utilizar uma técnica diferente para comparar os "melhores modelos selecionados"visto que o nosso objetivo é prever se um dado indivíduo tem diabetes ou não. Sendo uma previsão decidimos então dividir o *dataset* em dados de treino (75%) e teste (25%). para que ao comparar os dados conseguíssemos fazê-lo com dados *fresh*. Isto é importante, pois o melhor modelo de previsão é aquele que prevê melhor os dados novos consoante os dados de treino. Dado que no futuro a ideia é prever um conjunto vasto de novos indivíduos então decidimos comparar com um número mais significativo de dados de teste.

Selecionamos os seguintes modelos:

- Regressão Linear
  - glyhb stab.glu + age
  - glyhb stab.glu + ratio + age
- Regressão Logística
  - diabetesB $stab.glu^2$ + ratio
  - diabetesB $stab.glu^2$
- KNN
  - diabetesB stab.glu + age + bp.1s, com K=10
  - diabetesB stab.glu + age + bp.1s, com K=12

De referir que os dados de teste e treino são iguais para a comparação entre todos os modelos, para que possam ser devidamente comparados.

Para a comparação entre os modelos tivemos de criar uma função que passa os valores do modelo linear para a variável binária, consoante o critério especificado no início do relatório: caso o valor de **glyhb** seja superior a 7 então tem diabetes; caso contrário não tem.

De seguida apresentamos os resultados obtidos e o melhor modelo.

Tabela 5: Tabela com o resultado das seleções entre os melhores modelos

Método	Modelo	Threshold	Acerto	Acerto positivos	Acerto Negativos	Acerto Ponderado
LM	glyhb∼stab.glu+age	glyhb >7	0.9381	0.7143	0.9759	3.5665
LM	glyhb∼stab.glu+ratio+age	glyhb >7	0.9278	0.6429	0.9759	3.4744
GLM	diabetesB∼$stab.glu^2$+ratio	0.39	0.9485	0.7143	0.9880	3.5991
GLM	diabetesB∼$stab.glu^2$	0.39	0.9485	0.7143	0.9880	3.5991
KNN	diabetesB∼stab.glu+age+bp.1s	K = 10	0.9588	0.7143	1	3.6318
KNN	diabetesB∼stab.glu+age+bp.1s	K = 12	0.9588	0.7143	1	3.6318

4.2.1 Modelo vencedor

Após as diferentes análises e considerações referidas anteriormente, decidimos escolher como vencedor o seguinte modelo.

$$LM : Stab.glu + age$$

Escolhemos este modelo da regressão linear não apenas pelo seu desempenho, visto que apesar de ter sido ligeiramente melhor não era uma diferença significativa em relação aos melhores modelos de cada método. A principal razão para a sua escolha foi o facto de ser um modelo bastante simples e fácil de se interpretar. Esse facto foi bastante importante

para a nossa análise pois consideramos que é preferível um modelo bastante mais simples com uma performance ligeiramente inferior a um modelo bastante complexo que seja bastante difícil de interpretar com um desempenho ligeiramente superior.

$$Fórmula : 1.693 + 0.027 * stab.glu + 0.020 * age$$

$$AdjustedR2 : 56.23\%$$

$$Acerto - ponderado : 3.63$$

$$AcertoTotal : 95.88\% - AcertoPositivos : 71.43\% - AcertoNegativos : 100\%$$

Como podemos concluir pelos dados apresentados as situações em que o modelo falha mais é no diagnóstico de pacientes com um resultado positivo, no entanto, é bastante bom a prever diagnósticos negativos, onde acertou todas as previsões.

Anomalia Verificamos apenas uma anomalia nestas comparações visto que o *p-value* do **ratio** no modelo descrito na imagem é alto pelo que o preditor é insignificante, algo que nos escapou aquando da análise da **regressão logística** visto que com a validação cruzada não foi possível verificar esta falha e os resultados nos acertos eram parecidos. De referir que este modelo obtém piores resultados que o outro e mesmo que os resultados fossem parecidos não seria considerado.

```
> glm.fit1 = glm(diabetesB~poly(stab.glu,2)+ratio,data=diabetesTreino,family=binomial)
> summary(glm.fit1)

Call:
glm(formula = diabetesB ~ poly(stab.glu, 2) + ratio, family = binomial,
    data = diabetesTreino)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2667  -0.3257  -0.2201  -0.1533   2.9450

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -3.5585     0.7181  -4.956 7.20e-07 ***
poly(stab.glu, 2)1  29.2021     3.8508   7.583 3.37e-14 ***
poly(stab.glu, 2)2 -13.1015     3.1009  -4.225 2.39e-05 ***
ratio             0.1949     0.1366   1.427    0.154
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 254.37  on 291  degrees of freedom
Residual deviance: 124.49  on 288  degrees of freedom
  (11 observations deleted due to missingness)
AIC: 132.49

Number of Fisher Scoring iterations: 6
```

Figura 18: Summary do glm com o modelo diabetesB: $stab.glu^2$ + ratio

5 Discussão dos resultados

Após termos selecionado, aquele que para nós é, o melhor modelo procedemos à resposta para as perguntas anteriormente colocadas.

5.1 Respostas

Será que o colesterol/pressão arterial/tempo após refeição/fatores corporais afetam os diabetes? O colesterol (chol), pressão arterial (bp.1*), tempo após refeição (time.ppn) e alguns fatores corporais como peso (weight) e largura da anca (hip) afetam a variável resposta ligeiramente, mas o ganho ao inserir estes elementos no modelo não justifica o aumento de complexidade.

Qual fator corporal explica melhor o valor da diabete? Para responder a esta pergunta é necessário dividi-la em duas outras perguntas:

- Qual fator corporal explica melhor o valor da variável resposta binária?
- Qual fator corporal explica melhor o valor da variável resposta hemoglobina glicada (glyhb)?

Deste modo, o fator corporal que mais influencia a variável resposta hemoglobina glicada é o perímetro da anca enquanto que o fator corporal que mais influencia a variável resposta binária é o perímetro da cintura (waist).

Quais fatores influenciam mais o resultado final? Os fatores que influenciam mais o resultado do teste de diabetes, segundo o nosso teste da influência, são a glucose estabilizada (stab.glu) e a idade (age) e o tempo após refeição (time.ppn). Quando inserimos estes fatores nos modelos, apenas a glucose estabilizada e a idade mostraram melhorias significativas aos respetivos modelos. O tempo após refeição trazia algumas melhorias mas estas não se mostraram significativas.

De que forma os fatores selecionados para a explicação do resultado o influenciam? Como mencionado na resposta à pergunta anterior, os fatores selecionados para a explicação do resultado final foram a glucose estabilizada e a idade. Ambos influenciam **linearmente** e de forma **crescente** o resultado do teste à diabete.

Qual a probabilidade (ou confiança no resultado) de uma pessoa com as caraterísticas X (por exemplo colesterol=180, altura=175, peso=67, etc.) ter diabetes? Esta pergunta não pode ser respondida diretamente pelo facto de termos usado uma regressão linear e a mesma não nos retornar uma probabilidade, mas sim o valor expectável da **glyhb**. No entanto existem métodos que nos podem aproximar da resposta à pergunta, como por exemplo:

- Qual a confiança no resultado quando dizemos que uma pessoa tem diabetes?

Para isso basta alterar o nível de "confiança"do intervalo de previsão (dado pelo modelo escolhido) até que o valor mínimo desse intervalo seja 7, valor a partir do qual um indivíduo contrai diabetes. O resultado será o nível de confiança utilizado para a previsão.

Para verificar esse nível realizamos testes consoante os dois preditores considerados importantes pelo modelo: **stab.glu** e **age** (o que nos mostra o quão afastados estávamos na pergunta inicial, antes de realizarmos qualquer análise aos dados). De seguida apresentamos 2 tabelas com os resultados obtidos.

Tabela 6: Valores dos resultados variando a **stab.glu**

stab.glu	age	Confiança (%)
151	60	0.5
171	60	29.9
191	60	54.7
211	60	73.8
231	60	86.3

Tabela 7: Valores dos resultados variando a **age**

stab.glu	age	Confiança (%)
191	20	16.7
191	40	37
191	60	54.7
191	75	66

Numa tabela apenas variamos a idade enquanto que noutra apenas variamos a **stab.glu**, o que nos mostra que apesar do modelo ser linear a diferença relativamente aos níveis não o é. Para além disso, podemos verificar que a **stab.glu** altera bastante mais a confiança do que a idade. Podemos afirmar que o *range* é diferente nas duas variáveis (**stab.glu**=231-151=80 e **age=75-20=55**), no entanto também sabemos que a **stab.glu** tem um alcance bastante superior à idade, pelo que em termos percentuais não difere assim tanto, enquanto que os níveis de confiança diferem bastante.

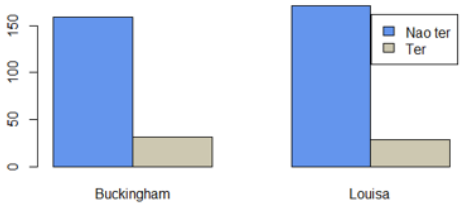
Em anexo seguem algumas imagens de mais resultados, naquele caso também para a probabilidade de não ter diabetes (a lógica é a mesma, mas em vez de se considerar o limite inferior considera-se o superior).

Qual a taxa de incidência em pessoas com menos e com mais de 50 anos? Para responder a esta pergunta realizamos apenas umas estatísticas relativamente aos dados e chegamos a estes resultados:

- 94.22% das pessoas que têm menos de 50 anos não têm diabetes
- 70,4% das pessoas que têm mais de 50 anos não têm diabetes
- 42.55% dos que não têm diabetes têm mais de 50 anos
- 83.54% dos que têm diabetes têm mais de 50 anos

Daqui podemos verificar que a incidência é superior em pessoas com mais de 50 anos, o que já nos é explicado pelo modelo visto que quanto maior a idade, maior o valor da **glyhb** e consequentemente maior a probabilidade de contrair diabetes.

Qual cidade apresenta maior incidência? (visto serem só dois podemos comparar) Como podemos ver na imagem abaixo ambas as cidades têm aproximadamente a mesma incidência. Isto já tinha sido concluído na nossa análise à influência, visto que a localização não afeta nenhuma das variáveis resposta.



xx

Figura 19: Diferenças da incidência entre as cidades

O resultado é mais exato utilizando um modelo de classificação ou de regressão (e de seguida classificando)? Esta pergunta foi quase como uma conclusão e resumo da nossa análise e seleção do modelo, visto que apenas temos de olhar para o modelo selecionado para responder à mesma. Tendo em conta esse modelo, podemos afirmar que o resultado é mais exato utilizando um modelo de regressão (e seguida classificação) do que o modelo de classificação.

Esta conclusão foi algo que já esperávamos antes de começar a realizar a seleção do modelo. No entanto isso mudou quando começamos a verificar que os *Adjusted R-squared* eram bastante baixos para a regressão linear, pelo que, por essa razão, nos acabou por surpreender um pouco.

6 Conclusão

Para terminar, gostaríamos apenas de referir que estamos satisfeitos com os resultados obtidos com este projeto. Como escolhemos um *dataset* menos "cru"para trabalharmos, visto que era um *dataset* preparado já para estes fins, decidimos dar um pouco menos importância à parte da análise exploratória e aumentar a incidência na parte dos modelos. Utilizamos uma quantidade considerável de modelos diferentes, com diferentes tipos de fórmulas, materializando todo o conhecimento obtido ao longo do semestre, o que para nós era sem dúvida o principal objetivo.

Obviamente que ficaram por fazer algumas coisas e algumas poderiam ter sido feitas de uma maneira mais correta, como a análise de diagnóstico, onde são processados os *outliers* e *leverage points*, que poderiam ter influenciado o resultado final de maneira diferente. Optamos por não realizar esta análise devido ao curto espaço de tempo e por não termos grandes informações de como foi efetuada a recolha dos dados que constituem o *dataset*. Para além disso poderíamos ter utilizado outros modelos, como o *shrinkage* (*lasso e ridge regression*), que talvez conduzissem a melhores resultados.

Anexos

Imagens

Análise de influência

Variável Binária

Primeiras 5

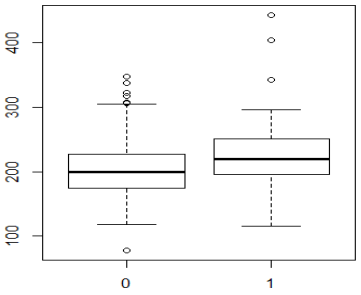


Figura 1 Colestrol

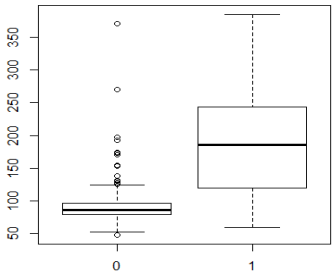


Figura 2 Stab.glu

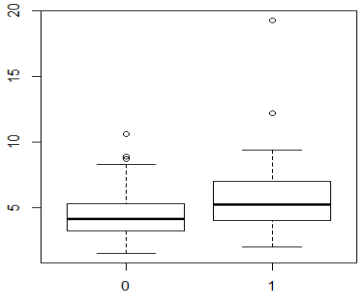


Figura 3 Ratio

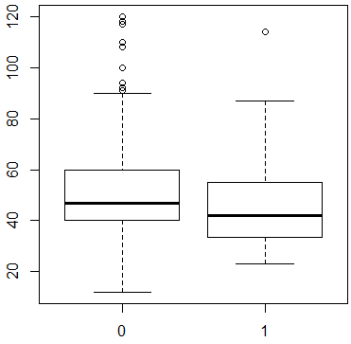


Figura 4 HDL

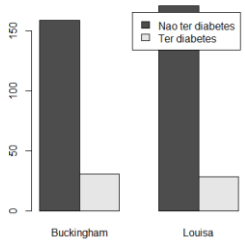
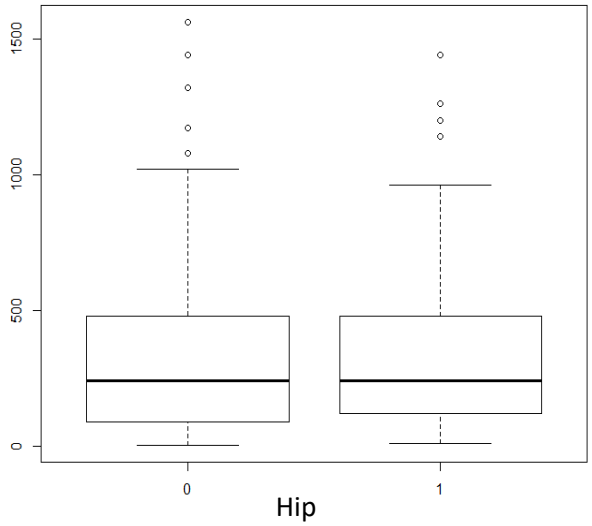


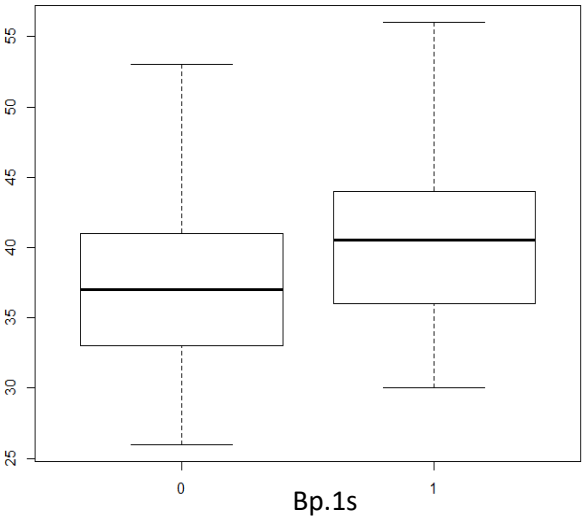
Figura 5 Location

5-10

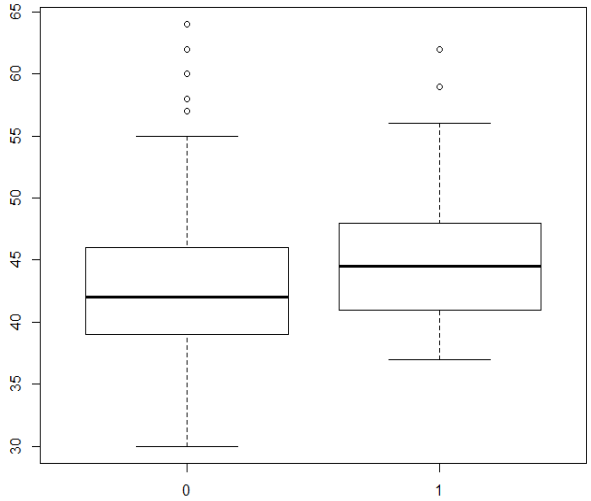
Time.ppn



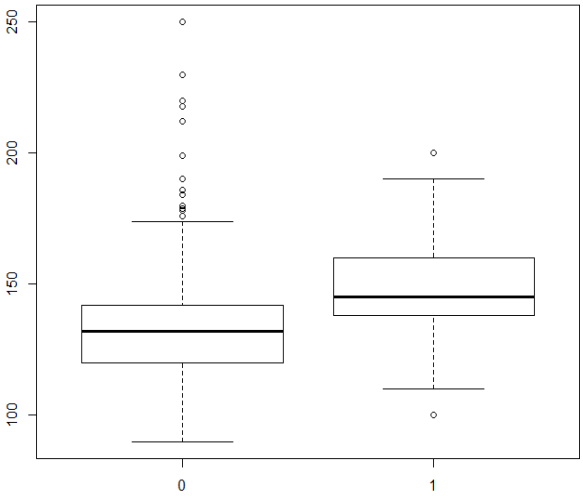
Waist



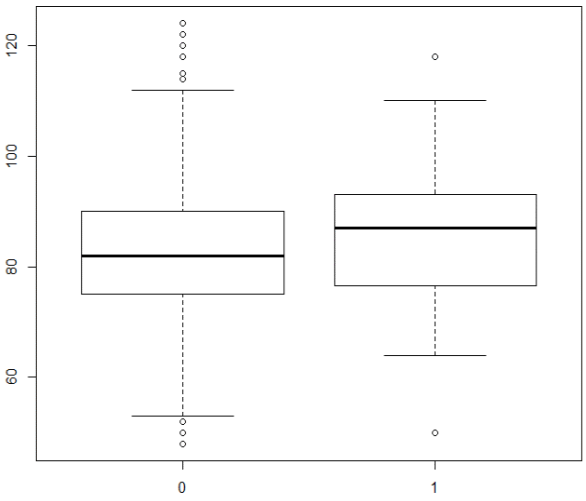
Hip



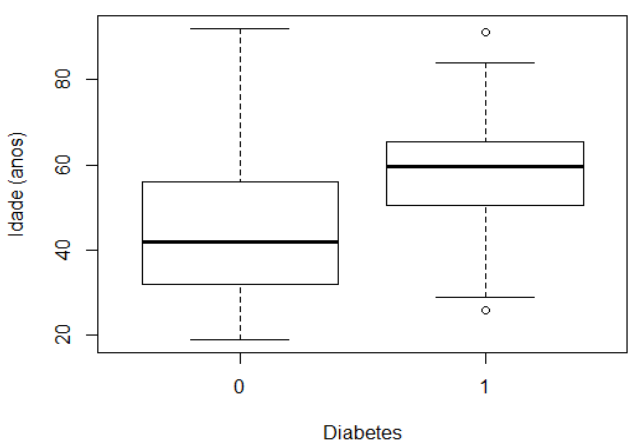
Bp.1s

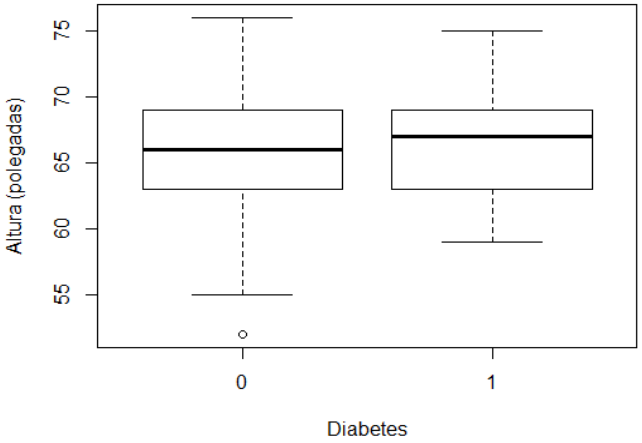


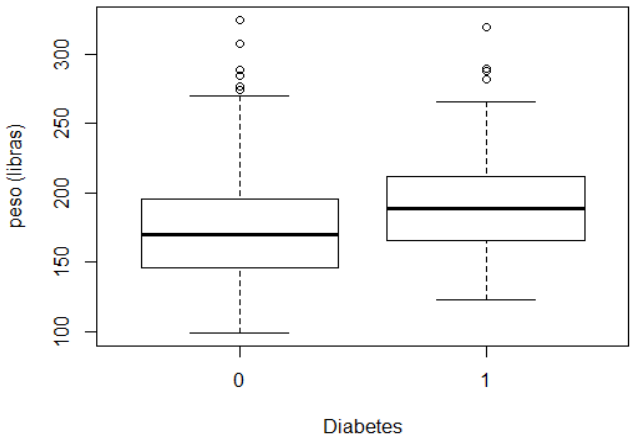
Bp.1d

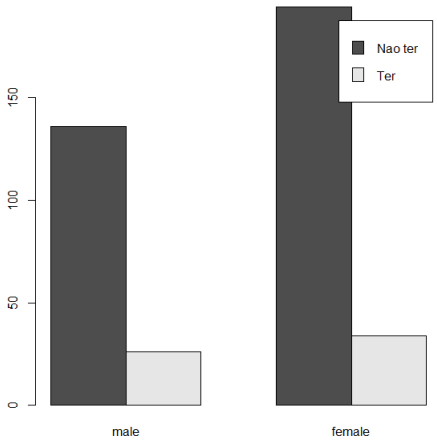


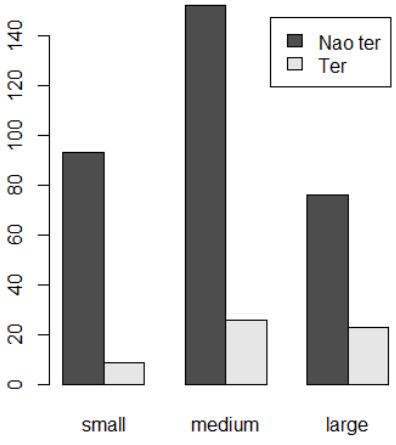
5-15


Idade (anos)
20
40
60
80
0
1
Diabetes

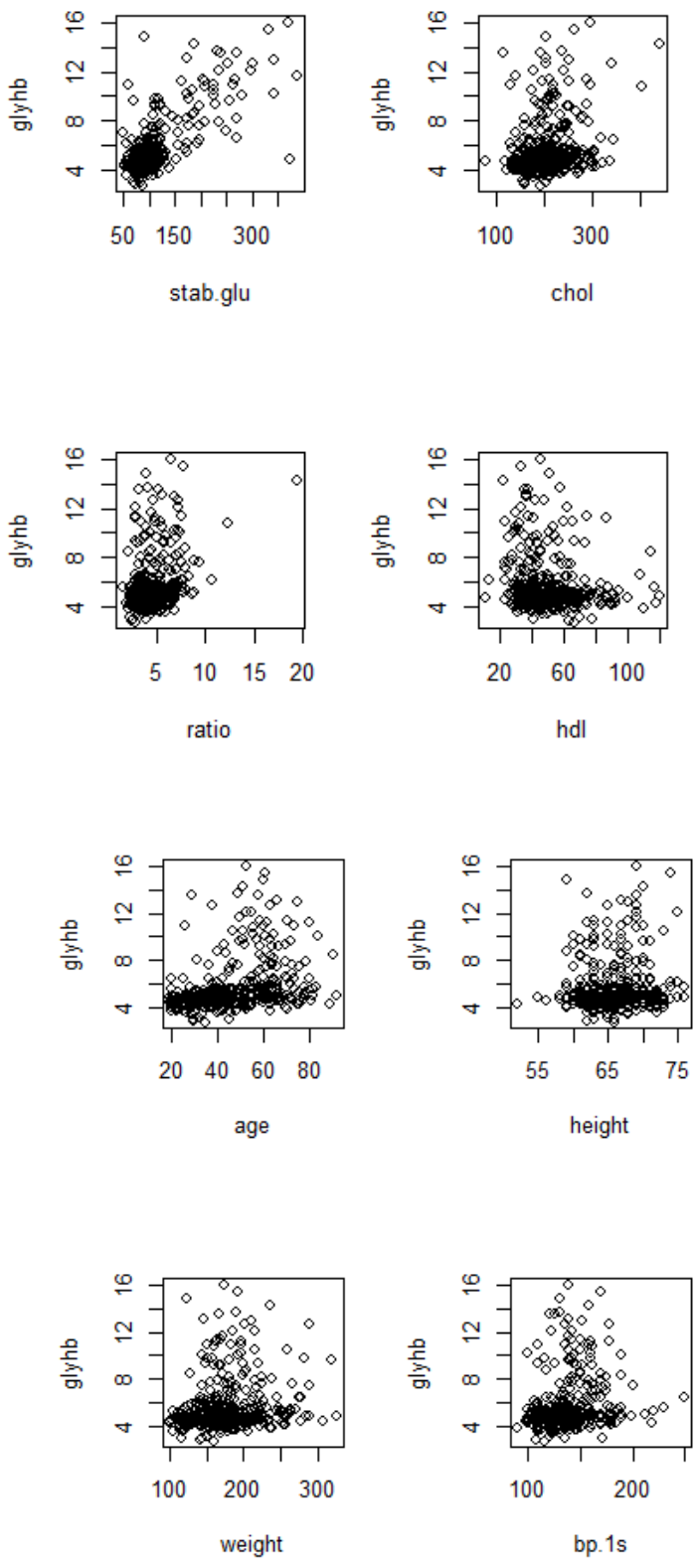

Altura (polegadas)
55
60
65
70
75
0
1
Diabetes

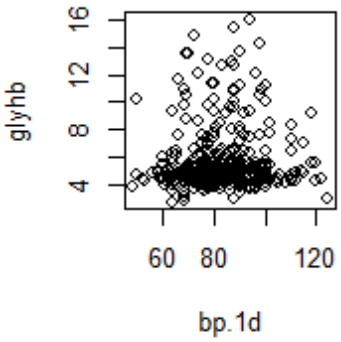

peso (libras)
100
150
200
250
300
0
1
Diabetes

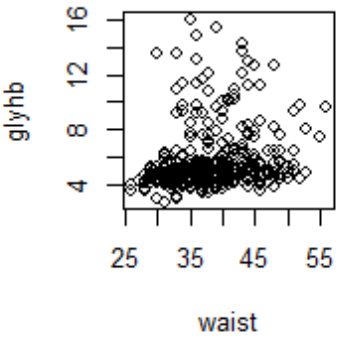

Nao ter
Ter
0
50
100
150
male
female

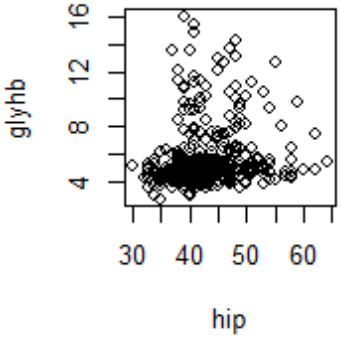

Nao ter
Ter
0
20
40
60
80
100
120
140
small
medium
large
Frame

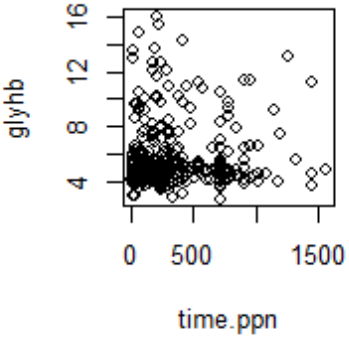
Variável quantitativa

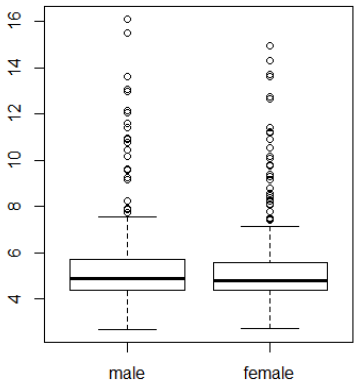




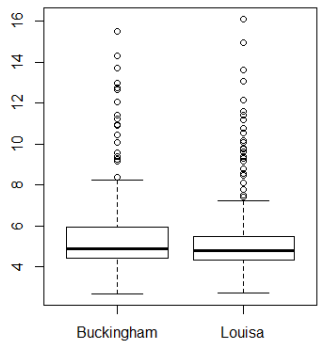




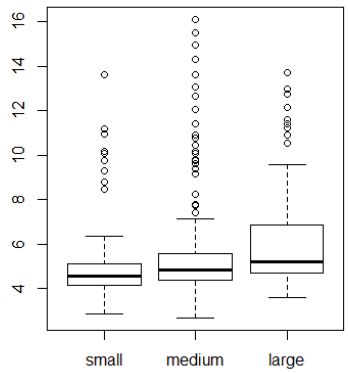




Gender



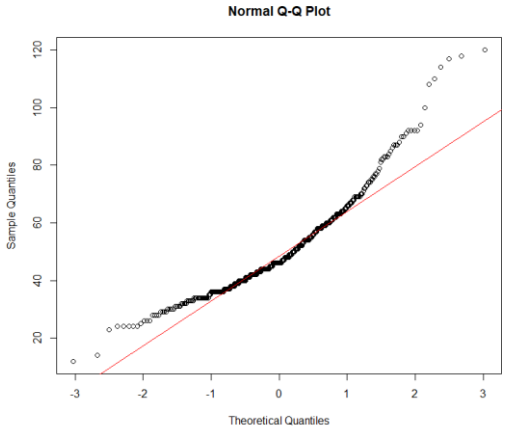
Location

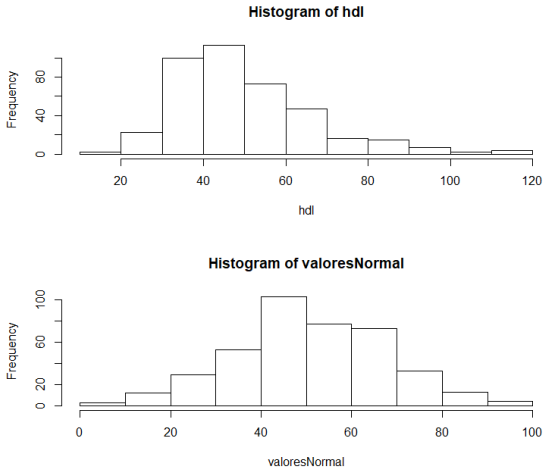


Frame

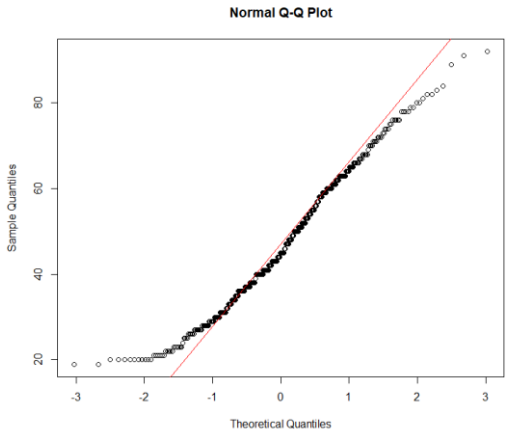
Análise à normalidade

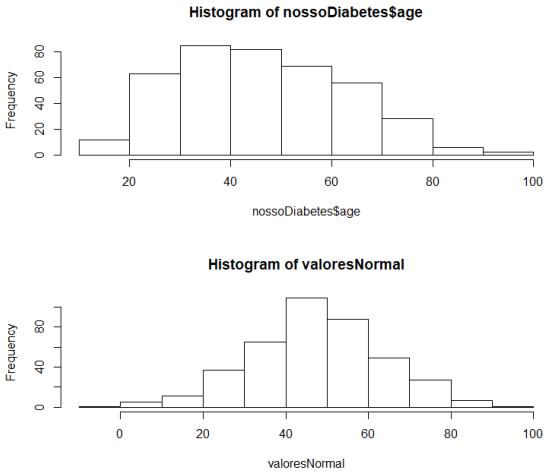
Lipoproteína de alta densidade – hdl



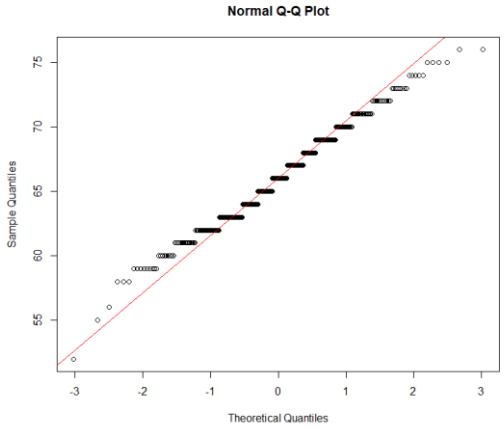


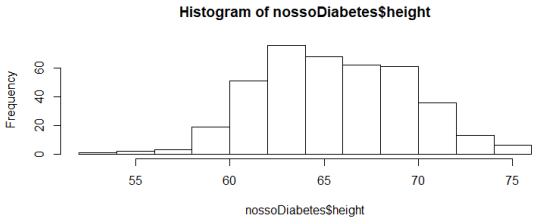
Idade – age

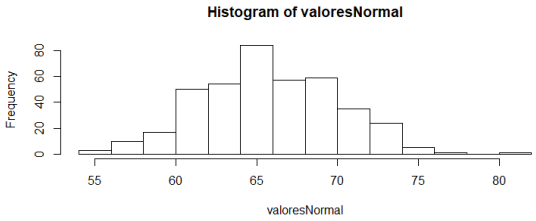




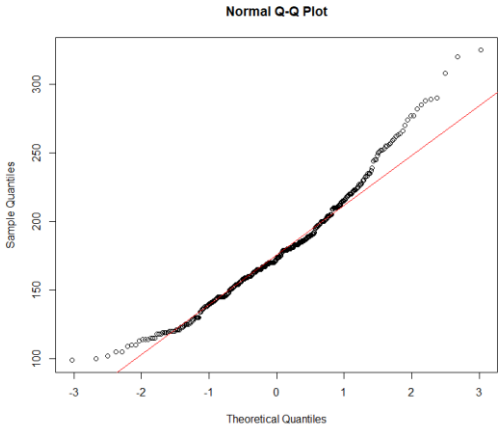
Altura – height

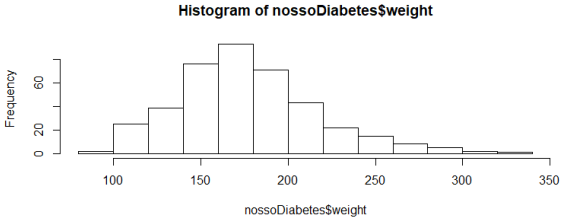


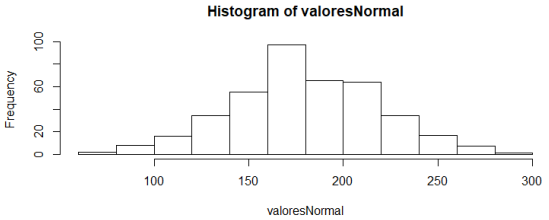




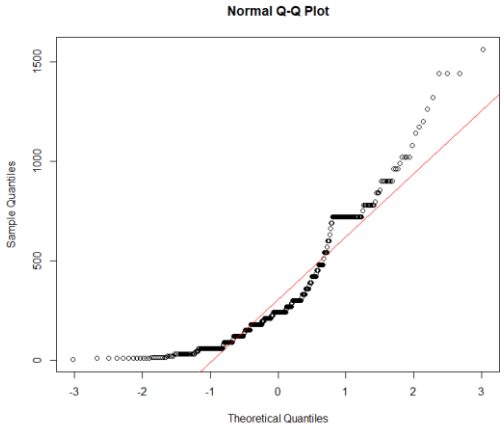
Peso – weight

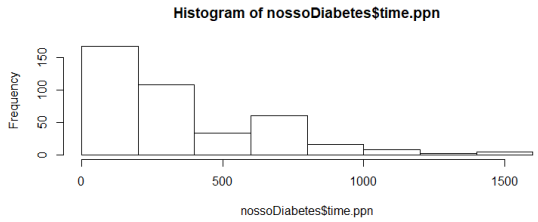


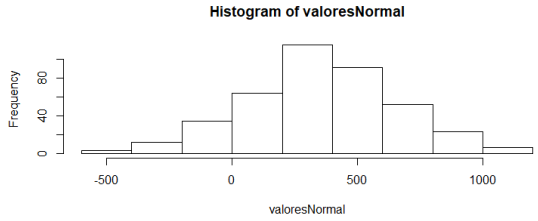




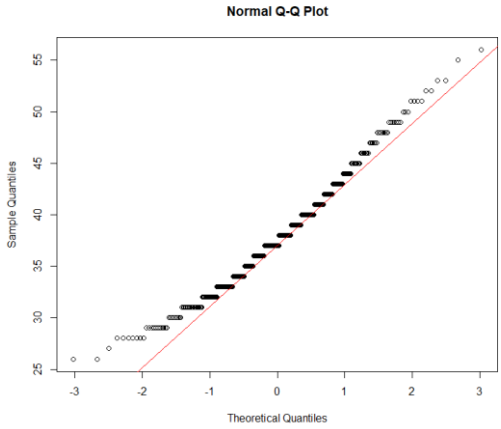
Tempo após refeição – time.ppn

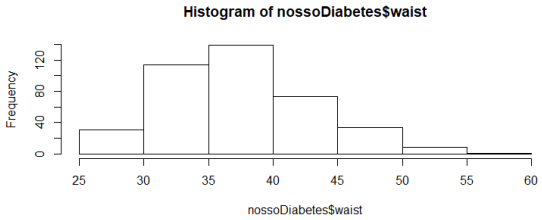


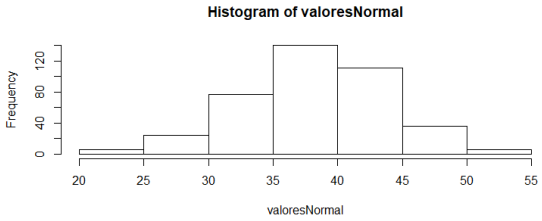




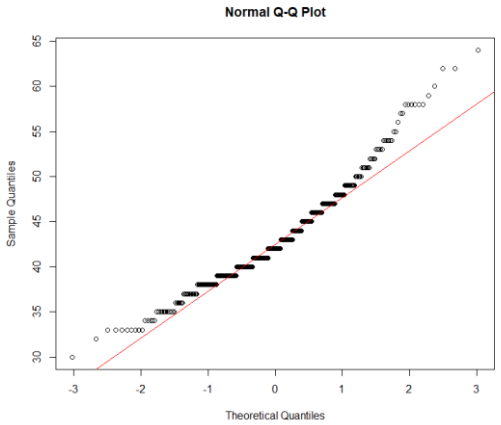
Perímetro da cintura – waist

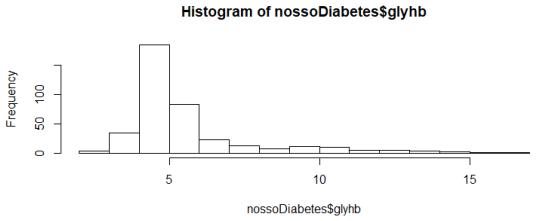


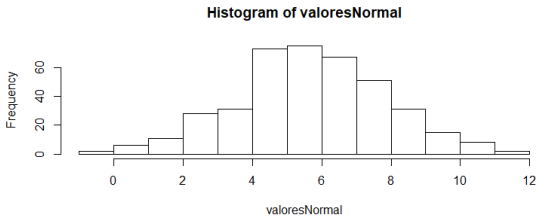




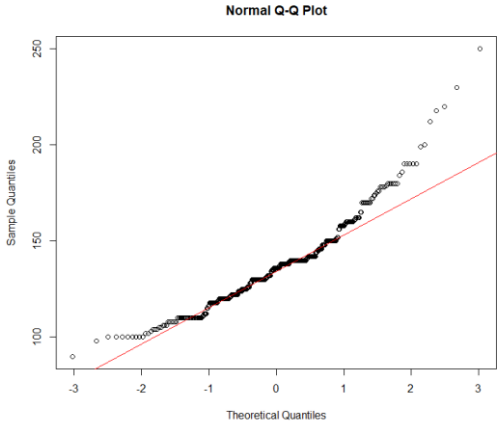
Perímetro da anca – hip

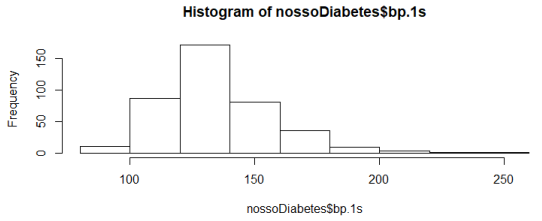


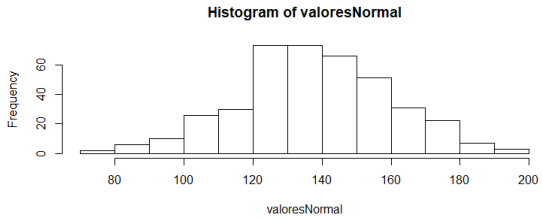




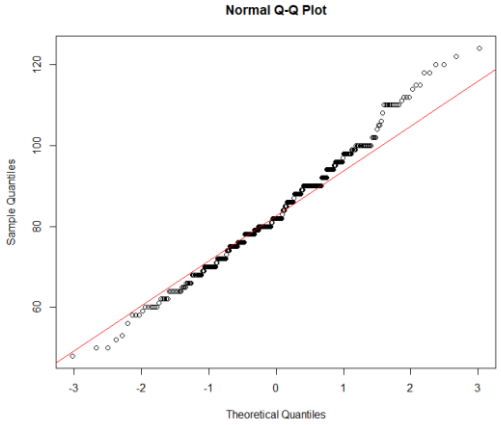
1º pressão arterial sistólica – bp.1s

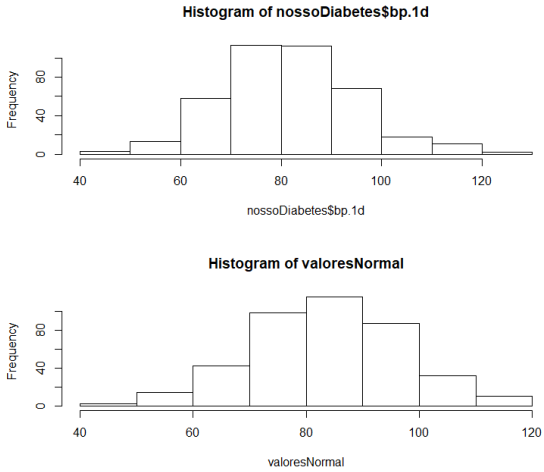






1º pressão arterial diastólica – bp.1d

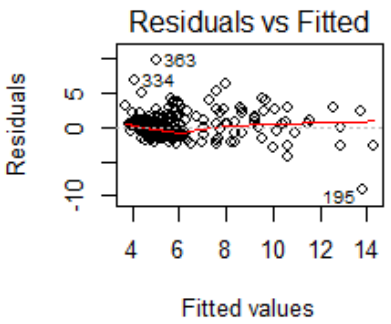


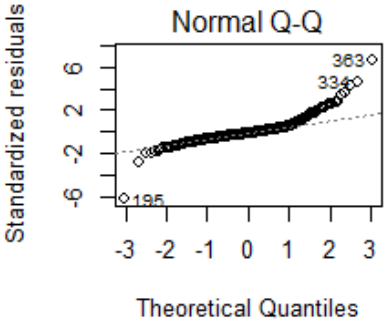


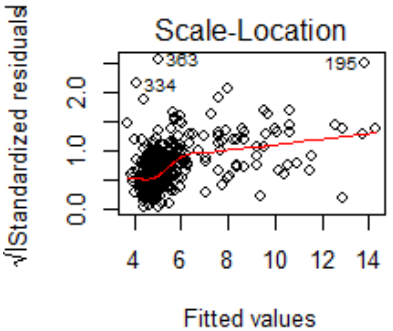
Imagens Regressão Linear

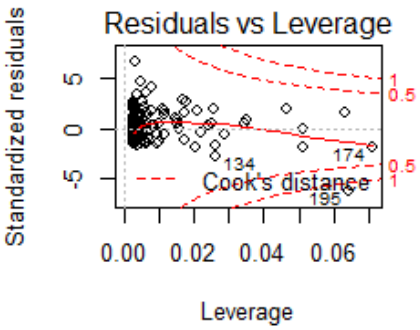
Nossa seleção

i) Stab.glu

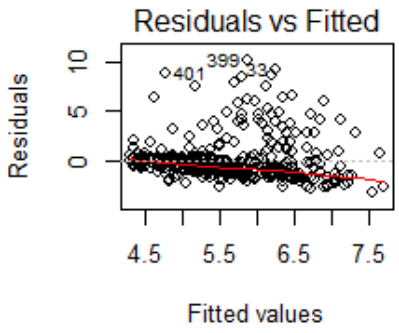


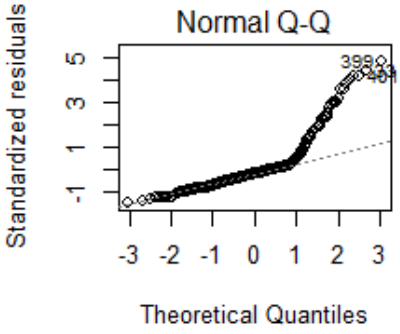


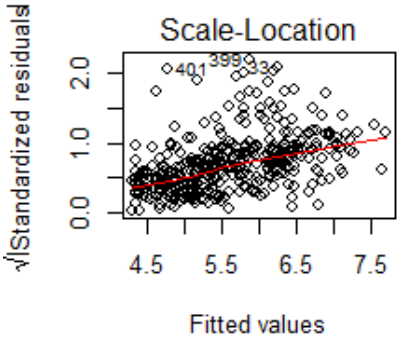


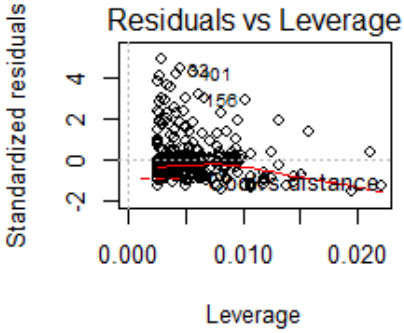


ii) Age

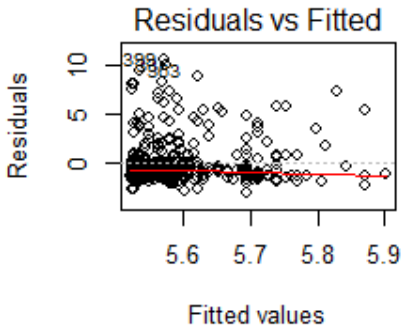


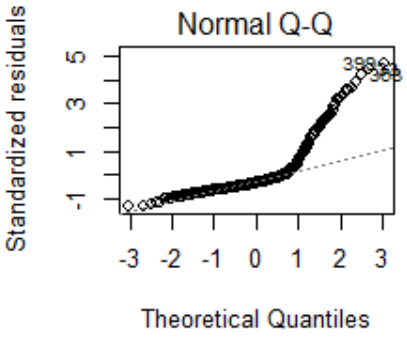


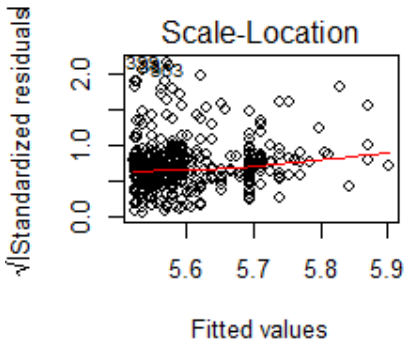


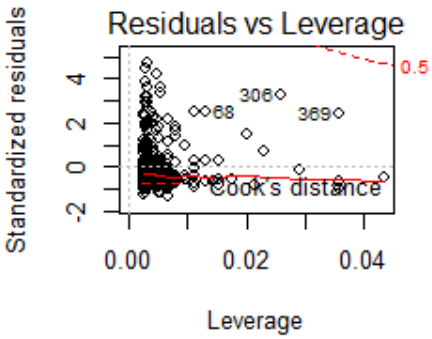


iii) Time.ppn

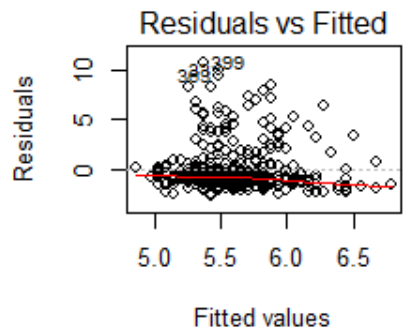


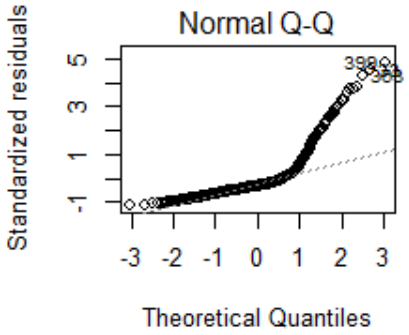


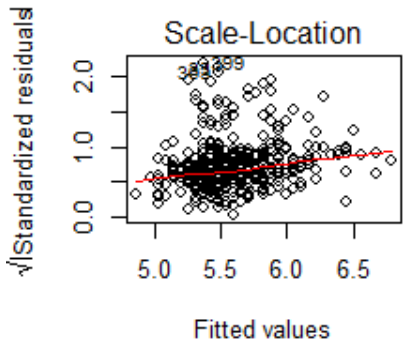


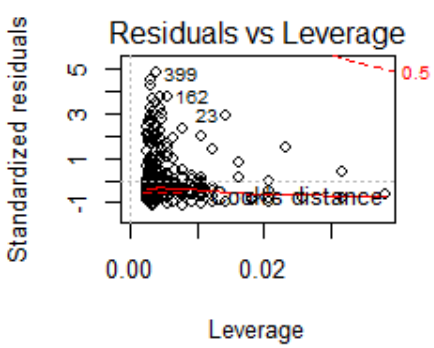


iv) Hip

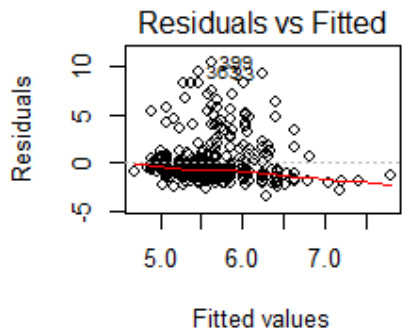

Residuals vs Fitted
Residuals
Fitted values
399

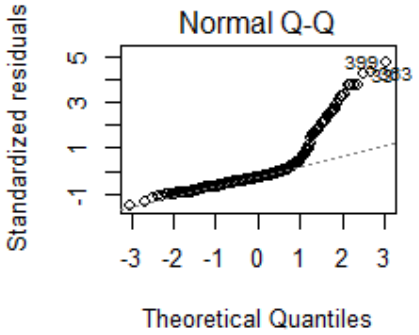

Normal Q-Q
Standardized residuals
Theoretical Quantiles
399

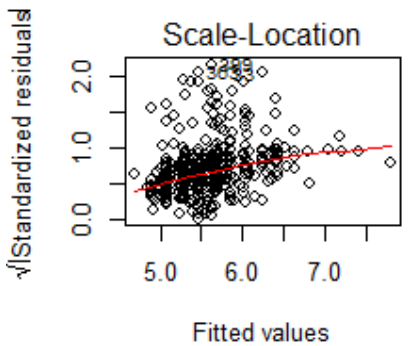

Scale-Location
√|Standardized residuals|
Fitted values
399

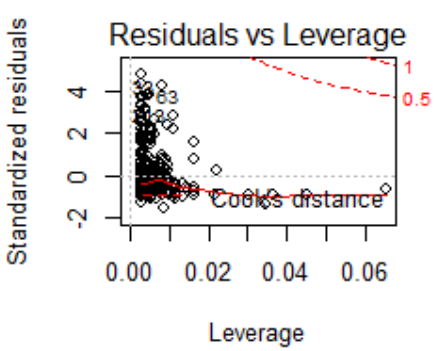

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
399
162
23
0.5

v) Bp.1s


Residuals vs Fitted
Residuals
Fitted values
399

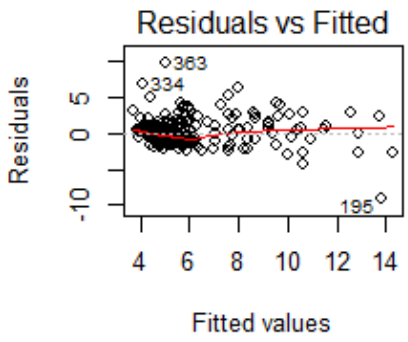

Normal Q-Q
Standardized residuals
Theoretical Quantiles
399

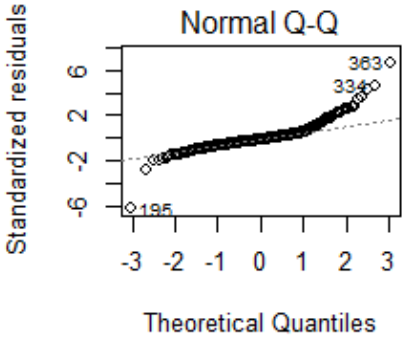

Scale-Location
√|Standardized residuals|
Fitted values
399

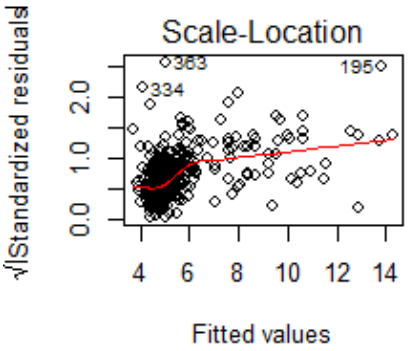

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
1
0.5

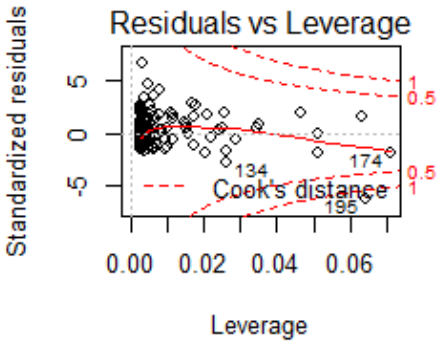
Seleção do regsubsets

i) Stab.glu

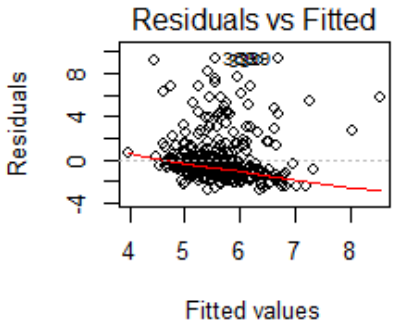


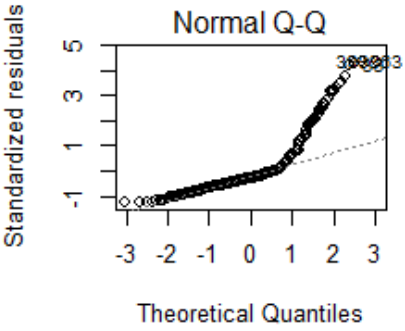


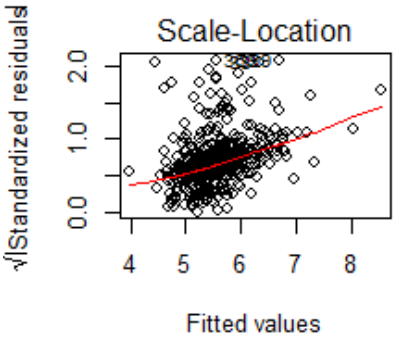


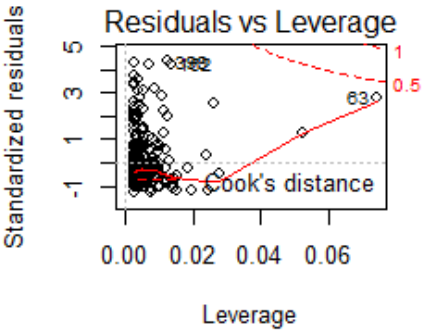


ii) Chol

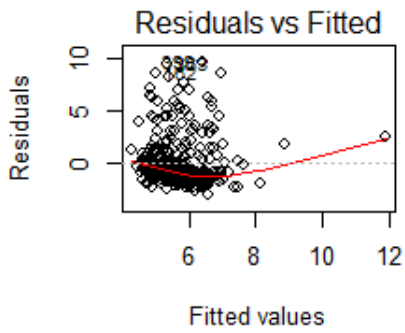


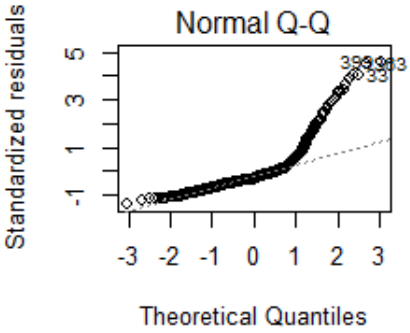


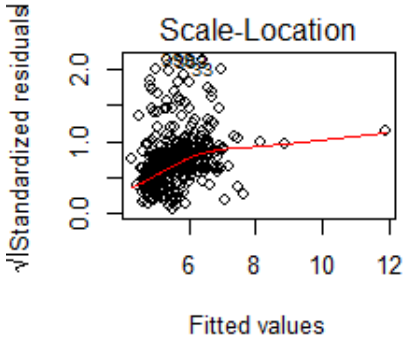


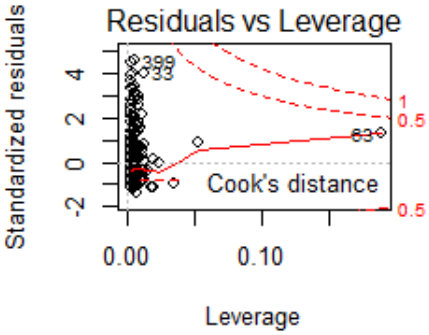


iii) Ratio

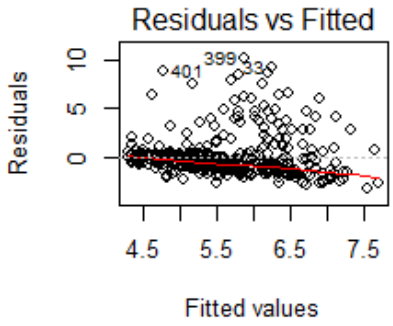


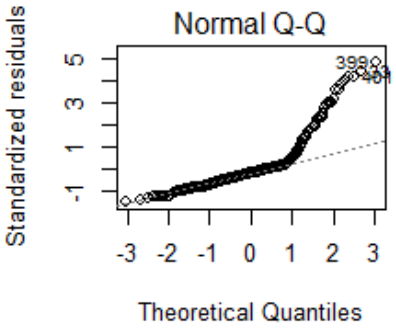


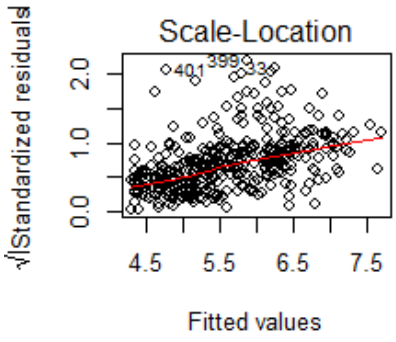


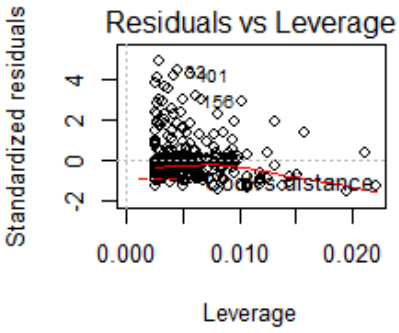


iv) Age

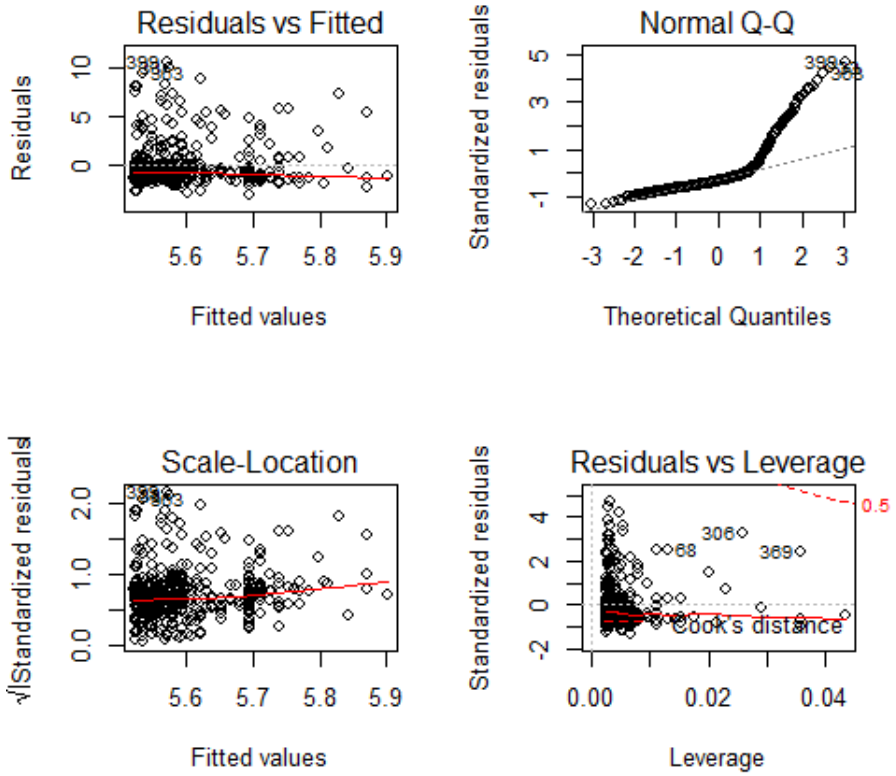








v) *Time.ppn*



Tabelas utilizadas para responder às questoes

Não ter diabetes

Stab.glu	Age	Confiança
91	15	91.1%
91	20	89.999%
91	50	78.5%
91	70	66.6%

Stab.glu	Age	Confiança
61	20	97.1%
91	20	89.999%
121	20	72.1%
151	20	40%

Ter diabetes:

Stab.glu	Age	Confiança
191	20	16.7%
191	40	37%
191	60	54.7%
191	75	66%

Stab.glu	Age	Confiança
151	60	0.5%
171	60	29.9%
191	60	54.7%
211	60	73.8%
231	60	86.3%

Código

```
library(faraway)
library(visdat)
library(corrplot)
library(MASS)
library(class)
library(dplyr)
library(ISLR)
library(leaps)
install.packages("caret")
library(caret)

?diabetes
vis_miss(diabetes[,-1])
## Retirar o id, visto n ser importante
diabetes
nossoDiabetes <- diabetes[,-1]
nossoDiabetes

## Gerar a variável binária que indica se tem ou não diabetes
diabetesB <- rep(NA,403)
diabetesB[nossoDiabetes$glyhb > 7]=1
diabetesB[nossoDiabetes$glyhb <= 7] = 0
# por como fator (é qualitativa)
diabetesB <- factor(diabetesB)
summary(diabetesB)

##Adicionar a variavel ao dataset
nossoDiabetes$diabetesB <- diabetesB

##Ordenar
nossoDiabetes <- nossoDiabetes[,c(19,5,1,2,3,4,6,7,8,9,10,11,12,13,16,17,18)]
summary(nossoDiabetes)

##Visualizar as correlações
cor(nossoDiabetes[,-c(1,7,9,12)],use = "complete.obs")[1,] #apenas da hemoglobina glicada
rel = cor(nossoDiabetes[,-c(1,7,9,12)],use = "complete.obs")
corrplot(rel, type="upper", tl.col = "black",sig.level = 0.01)
```

Análise Exploratória

Análise à influência

```
#---------------------#
# Teste à influência |
#---------------------#

##Gerar os gráficas das variáveis em função das outras
pairs(nossoDiabetes[,c(1:6)])
pairs(nossoDiabetes[,c(1:2,7:10)])
pairs(nossoDiabetes[,c(1:2,11:14)])
pairs(nossoDiabetes[,c(1:2,15:19)])


### Variável Binária:
par(mfrow=c(1,1))

### ANLISE COLESTROL ##
```

```
colestrol <- diabetes$chol
table(diabetesB)
table(diabetesB,colestrol)
table(diabetesB,colestrol)
boxplot(table(diabetesB,colestrol))
boxplot(colestrol~diabetesB)
barplot(colestrol~diabetesB) ##Pode ter inflcuencia maior o colestrol diabetes bitch, mediana ligeiramente a
    cima
table(diabetesB)
fivenum(colestrol)
#Min: 78, Max:443, Media 179, 25%->78 75%->230

### ANLISE STAB-GLU ##
#FIVENUM()
stab = nossoDiabetes$stab.glu
fivenum(stab)
#Min: 48, Max:385, Media 89, 25%->81 75%->106
table(diabetesB,stab)
boxplot(stab)
boxplot(stab~diabetesB) #STAB-GLU CLARAMENTE AFETA

##ANALISE RACIO ##
racio <- diabetes$ratio
summary(racio)
boxplot(racio~diabetesB) ## Parece ter algumo

### ANLISE HDL ##
hdl <- diabetes$hdl
table(diabetesB,hdl)
fivenum(hdl)
#Min: 12, Max:120, Media 46, 25%->38 75%->59
boxplot(hdl)
boxplot(hdl~diabetesB) ##Mais baixo HDL menor os "diabetes"

### ANLISE location ##
loc <- diabetes$location
table(diabete,loc)
barplot(table(diabetesB,loc),beside=T, legend.text=c("Nao ter","Ter")) ## Nao parece ter influencia nenhuma
    a localizao

##ANALISE AGE
age <- diabetes$age
summary(age)
table(diabetesB,age)
barplot(table(diabetesB,age))
boxplot(age)
boxplot(age~diabetesB)


# Age
boxplot(nossoDiabetes$age~nossoDiabetes$diabetesB) ## parece afetar um pouco
# Height
boxplot(nossoDiabetes$height~nossoDiabetes$diabetesB) ## não parece afetar praticamente nada
# Weight
boxplot(nossoDiabetes$weight~nossoDiabetes$diabetesB) ## parece afetar um pouco (menos que
## a primeira)

# Gender
table(nossoDiabetes$diabetesB, nossoDiabetes$gender) ## Na tabela podemos verificar que 18.2% dos
```

```
26/162 #1 em homens ## homens do dataset possuem diabetes. Já nas
34/228 #1 em mulheres ## mulheres a percentagem é de 17%. Tendo isto

136/(136+194) #homens em 0 ## conta o género parece não afetar
26/(26+34) #homens em 1

# Frame
table(nossoDiabetes$diabetesB, nossoDiabetes$frame) ## Na tabela podemos verificar que quanto maior
9/102 #1 em small ## é o frame, mais incidências de diabetes
26/178 #1 em medium ## são registadas. Parece portanto afetar
23/99 #1 em large

93/(93+152+76) #small em 0
152/(93+152+76) #medium em 0

9/(9+26+23) #small em 1
26/(9+26+23) #medium em 1
par(mfrow=c(1,1))

summary(nossoDiabetes$diabetesB)
boxplot(nossoDiabetes$time.ppn~nossoDiabetes$diabetesB) ##não parece afetar a
##variavel binaria
boxplot(nossoDiabetes$waist~nossoDiabetes$diabetesB) #parece afetar ligeiramente
boxplot(nossoDiabetes$hip~nossoDiabetes$diabetesB) ##parece afetar ligeiramente
boxplot(nossoDiabetes$bp.1s~nossoDiabetes$diabetesB) ##parece afetar mais, mas
## nao muito
boxplot(nossoDiabetes$bp.1d~nossoDiabetes$diabetesB) ##parece afetar ligeiramente (mas
## parece afetar mais do que a primeira, menos do que as restantes)

### Variável glyhb:
# Gender
boxplot(nossoDiabetes$glyhb~nossoDiabetes$gender) ## não parece afetar praticamente nada

# Frame
boxplot(nossoDiabetes$glyhb~nossoDiabetes$frame) ## não parece afetar praticamente nada

# Location
boxplot(nossoDiabetes$glyhb~nossoDiabetes$location) ## não parece afetar praticamente nada

plot(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn, type = "p") ##parece n afetar
abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn)) ##tendencia constante
plot(nossoDiabetes$glyhb ~ nossoDiabetes$waist, type = "p") ##dificil de
##analisar este grafico, no entanto ao fazer o fit parece ter tendencia
abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$waist)) ##crescente
plot(nossoDiabetes$glyhb ~ nossoDiabetes$hip, type = "p") ##o mesmo do anterior
abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$hip))
plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1s, type = "p")##parece afetar com
##tendencia crescente
abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1s)) ##confirma
plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d, type = "p")##nao parece afetar
abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d)) ##tendencia constante

par(mfrow=c(2,2))
plot(nossoDiabetes$glyhb ~ nossoDiabetes$chol, type = "p")
plot(nossoDiabetes$glyhb ~ nossoDiabetes$stab.glu, type = "p")
plot(nossoDiabetes$glyhb ~ nossoDiabetes$hdl, type = "p")
plot(nossoDiabetes$glyhb ~ nossoDiabetes$ratio, type = "p")
plot(nossoDiabetes$glyhb ~ nossoDiabetes$age, type = "p")
plot(nossoDiabetes$glyhb ~ nossoDiabetes$height, type = "p")
```

```
132 plot(nossoDiabetes$glyhb ~ nossoDiabetes$weight, type = "p")
133 plot(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn, type = "p")
134 plot(nossoDiabetes$glyhb ~ nossoDiabetes$waist, type = "p")
135 plot(nossoDiabetes$glyhb ~ nossoDiabetes$hip, type = "p")
136 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1s, type = "p")
137 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d, type = "p")
```

Regsubsets

```
1  par(mfrow=c(1,1))
2  selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16)
3  summary(selecao)
4  plot(selecao,scale="adjr2")
5
6
7  selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "forward")
8  summary(selecao)
9  plot(selecao,scale="adjr2")
10
11 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "backward")
12 summary(selecao)
13 plot(selecao,scale="adjr2")
14
15 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "seqrep")
16 summary(selecao)
17 plot(selecao,scale="adjr2")
```

Análise à normalidade

```
1  #---------------------#
2  # Teste à normalidade |
3  #---------------------#
4
5
6  ### ANLISE COLESTROL ##
7  colestrol <- diabetes$chol
8  qqnorm(colestrol) ##Parece normal ao contrario do ze
9  qqline(colestrol, col = 2)
10 hist(colestrol)
11 shapiro.test(colestrol)
12 valoresNormal <- rnorm(400,mean=mean(colestrol, na.rm = TRUE), sd=sd(colestrol, na.rm = TRUE))
13 hist(valoresNormal) ##PARECE NORMAL
14 fivenum(colestrol)
15 #Min: 78, Max:443, Media 179, 25%->78 75%->230
16
17
18
19 ### ANLISE STAB-GLU ##
20 stab <- diabetes$stab.glu
21 hist(stab) #
22 #FIVENUM()
23 fivenum(stab)
24 #Min: 48, Max:385, Media 89, 25%->81 75%->106
25 valoresNormal <- rnorm(400,mean=mean(stab, na.rm = TRUE), sd=sd(stab, na.rm = TRUE))
26 hist(valoresNormal) ##PARECE +- NORMAL
27
28 ##ANALISE RACIO ##
29 racio <- diabetes$ratio
30 hist(racio)
31 summary(racio)
```

```
32 valoresNormal <- rnorm(400,mean=mean(racio, na.rm = TRUE), sd=sd(racio, na.rm = TRUE))
33 hist(valoresNormal)
34 qqnorm(racio)
35 qqline(racio, col = 2) ##Dificil de analisar
36
37
38
39 ### ANLISE HDL ##
40
41 hdl <- diabetes$hdl
42 #Histograma
43 hist(hdl)
44 valoresNormal <- rnorm(400,mean=mean(hdl, na.rm = TRUE), sd=sd(hdl, na.rm = TRUE))
45 hist(valoresNormal) ##PARECE +- NORMAL
46 fivenum(hdl)
47 #Min: 12, Max:120, Media 46, 25%->38 75%->59
48
49
50 ##ANALISE AGE
51 age <- diabetes$age
52 summary(age)
53 hist(age)
54 qqnorm(age)
55 qqline(age, col="2")
56 valoresNormal <- rnorm(400,mean=mean(age, na.rm = TRUE), sd=sd(age, na.rm = TRUE)) ## Dificil de analisar
57 hist(valoresNormal) ##PARECE NORMAL
58
59
60 # Age
61
62 par(mfrow=c(2,1))
63 hist(nossoDiabetes$age) ## Aproxima-se da distribuição normal, contudo do lado esquedo apresenta
64 ## um crescimento bastante mais acentuado que do lado direito.
65 mean(nossoDiabetes$age,na.rm=TRUE)
66 sd(nossoDiabetes$age,na.rm=TRUE)
67 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$age, na.rm = TRUE), sd=sd(nossoDiabetes$age, na.rm = TRUE
       ))
68 hist(valoresNormal) ## Observando os dois diagramas podemos afirmar que são parecidos. è possivel
69 ## notar valores ligeiramente mais elevados à esquerda da média no de cima
70 summary(nossoDiabetes$age)
71 shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto pelo que não se rejeita
72 shapiro.test(nossoDiabetes$age) ## O diagrama superior tem um pvalor muito reduzido pelo que se
73 ## rejeita a hipótese nula (H0: dados normalmente distribuídos)
74 par(mfrow=c(1,1))
75 qqnorm(nossoDiabetes$age)
76 qqline(nossoDiabetes$age, col = 2) ## Do gráfico resultante podemos ver que a variável "Age"
77 ## não parece ser normalmente distribuida. Apesar de entre os
78 ## 35 e os 45 anos mostrar bastante convergência, fora deste
79 ## intrevalo é os valores divergem bastante
80
81 # Height
82 par(mfrow=c(2,1))
83 hist(nossoDiabetes$height) ## Verifica-se algumas semelhanças à distribuição normal (maior
84 ## probabilidade em torno do centro), contudo é possivel verificar
85 ## que na zona central a frequência mostra-se relativamente constante
86 ## ao longo de um largo intrevalo.
87 mean(nossoDiabetes$height,na.rm=TRUE)
88 sd(nossoDiabetes$height,na.rm=TRUE)
89 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$height, na.rm = TRUE), sd=sd(nossoDiabetes$height, na.rm
```

```
    = TRUE))
hist(valoresNormal) ## Observando os dois diagramas pé possível verificar algumas diferenças com
## as freqências mais elevadas a mostrarem-se em torno das 67 polegadas no
## diagrama inferior ao passo que no diagrama superior as frequências mais
## elevadas se encontravam em torno das 63 polegadas
summary(nossoDiabetes$height)
shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto (30%) pelo que não se rejeita
shapiro.test(nossoDiabetes$height) ## O diagrama superior tem um pvalor muito reduzido (0.2%) pelo
## que se rejeita a hipótese nula.
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$height)
qqline(nossoDiabetes$height, col = 2) ## Do gráfico resultante podemos ver que a variável "Height"
## não parece ser normalmente distribuida. Para alturas
## inferiores a 62 e superiores a 72 polegadas verifica-se
## uma tendência a divergir dos valores teóricos.


# Weight
set.seed(1)
par(mfrow=c(2,1))
hist(nossoDiabetes$weight) ## Olhando para o histograma é possível retirar bastantes parecências a
## uma distribuição normal. Do lado esquerda dos 170 pounds verifica-se
## uma subida mais constante da frequência relativamente ao lado direito
## (os dois lados diferem ligeiramente)
mean(nossoDiabetes$weight,na.rm=TRUE)
sd(nossoDiabetes$weight,na.rm=TRUE)
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$weight, na.rm = TRUE), sd=sd(nossoDiabetes$weight, na.rm
    = TRUE))
hist(valoresNormal) ## Observando os dois diagramas é possível verificar bastantes semelhanças
## entre eles nomeadamente no valor onde a frequência é mais elevada (170 pounds)
## e também a nível da distribuição à esquerda e à direita deste valor.
summary(nossoDiabetes$weight)
shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto (80%) pelo que não se rejeita
shapiro.test(nossoDiabetes$weight) ## O diagrama superior tem um pvalor muito reduzido (<0.01%) pelo
## que se rejeita a hipótese nula.
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$weight)
qqline(nossoDiabetes$weight, col = 2) ## Do gráfico resultante podemos ver que a variável "Weight"
## não parece ser normalmente distribuida. Para pesos
## inferiores a 130 pounds e superiores a 220 pounds
## verifica-se uma tendência a divergir superiormente dos
## valores teóricos.


par(mfrow=c(2,1))
hist(nossoDiabetes$glyhb) ##tem algumas parecencas com distribuição normal
## no entanto tem varios valores para o lado de lá (apesar de serem muito poucos)
## o melhor será verificar com uma linha da distribuição normal
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$glyhb, na.rm = TRUE), sd=sd(nossoDiabetes$glyhb, na.rm =
    TRUE))
hist(valoresNormal)
summary(nossoDiabetes$glyhb)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$glyhb) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
par(mfrow=c(1,1))
qqnorm(valoresNormal)
qqline(valoresNormal, col = 2)
qqnorm(nossoDiabetes$glyhb)
```

```
qqline(nossoDiabetes$glyhb, col = 2) ##do grafico do teste da normal vemos que
##a variavel se afasta muito e não parece normalmente distribuida


par(mfrow=c(2,1))
hist(nossoDiabetes$time.ppn) ##não se aproxima da distribuição normal
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$time.ppn, na.rm = TRUE), sd=sd(nossoDiabetes$time.ppn, na
    .rm = TRUE))
hist(valoresNormal)
summary(nossoDiabetes$time.ppn)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$time.ppn) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$time.ppn)
qqline(nossoDiabetes$time.ppn, col = 2) ##do grafico do teste da normal vemos que
##a variavel se afasta muito e não parece normalmente distribuida


par(mfrow=c(2,1))
hist(nossoDiabetes$waist) ##aproxima-se algo da distribuição normal
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$waist, na.rm = TRUE), sd=sd(nossoDiabetes$waist, na.rm =
    TRUE))
hist(valoresNormal) ##os histogramas são pareceidos
summary(nossoDiabetes$waist)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$waist) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
par(mfrow=c(1,1))
qqnorm(valoresNormal)
qqline(valoresNormal, col = 2)
qqnorm(nossoDiabetes$waist)
qqline(nossoDiabetes$waist, col = 2) ##do grafico do teste da normal vemos que
##a variavel nao se afasta muito e parece normalmente distribuida (no meio
## até se comporta bem, mas nos extremos começa a afastar um pouco embora
## aparenta afastar-se menosdo que o seguinte, nomeadamente no extremo superior)


par(mfrow=c(2,1))
hist(nossoDiabetes$hip) ##aproxma-se ligeiramente da distribuição normal
mean(nossoDiabetes$hip,na.rm=TRUE)
sd(nossoDiabetes$hip,na.rm=TRUE)
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$hip, na.rm = TRUE), sd=sd(nossoDiabetes$hip, na.rm = TRUE
    ))
hist(valoresNormal) ##comparando os 2 histogramas vemos que é parecido
summary(nossoDiabetes$hip)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$hip) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$hip)
qqline(nossoDiabetes$hip, col = 2) ##do grafico do teste da normal vemos que
##a variavel se afasta muito e não parece normalmente distribuida (no meio
## até se comporta bem, mas nos extremos começa a afastar muito, embora
## aparenta ser menos do queo seguinte)


par(mfrow=c(2,1))
hist(nossoDiabetes$bp.1s) ##tem algumas parecencas, mas tem o mm problema da
```

```
##primeira, embora com menos amplitude para lá (diferença do maior valor)
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$bp.1s, na.rm = TRUE), sd=sd(nossoDiabetes$bp.1s, na.rm =
    TRUE))
hist(valoresNormal) ##não são parecidos os histogramas, as freqs sao diferentes
summary(nossoDiabetes$bp.1s)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$bp.1s) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$bp.1s)
qqline(nossoDiabetes$bp.1s, col = 2) ##do grafico do teste da normal vemos que
##a variavel se afasta muito e não parece normalmente distribuida (no meio
## até se comporta bem, mas nos extremos começa a afastar muito)


par(mfrow=c(2,1))
hist(nossoDiabetes$bp.1d) ##parece normalmente distribuído
valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$bp.1d, na.rm = TRUE), sd=sd(nossoDiabetes$bp.1d, na.rm =
    TRUE))
hist(valoresNormal)
summary(nossoDiabetes$bp.1d)
shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
shapiro.test(nossoDiabetes$bp.1d) ##este teste tem um pvalor baixo pelo que se
##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
##no entanto este p-valor é bastante superior a todos os outros sendo de 0.01
par(mfrow=c(1,1))
qqnorm(nossoDiabetes$bp.1d)
qqline(nossoDiabetes$bp.1d, col = 2) ##do grafico do teste da normal vemos que
##a variavel se afasta um pouco, mas não muito pelo que não conseguimos
## concluir nada em concreto (aparenta ser normal)
```

Seleção do Modelo

Regressão Linear

```

## NOSSA ANALISE ##

## Teste com 1 var do regsubsets ## stab.glu
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu)
treino = treino[complete.cases(treino),]

lm.fit.Stab.glu = lm(glyhb~stab.glu, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Stab.glu)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos estão
# espalhados em volta de uma linha horizontal, o que indica que em princípio não
# existem relações não lineares
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos parecem
# seguir uma distribuição normal

lm.fit1 <- lm.fit.Stab.glu
summary(lm.fit1) # Adjusted R-squared: 0.5602

## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
```

```r
model1 <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
print(model1)
# Adjusted R-squared: 0.567815
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)

#------------------------------------------------------------------------------

# 2 variáveis -> stab.glu + age
set.seed(1)
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age)
treino = treino[complete.cases(treino),]

lm.fit.Age = lm(glyhb~age, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Age)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Age) # Adjusted R-squared: 0.1128
# Parece adicionar precisão ao modelo

lm.fitAnt <- lm(glyhb~stab.glu, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5602

lm.fit1 <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5745 -> Melhorou um pouco

# F-test parcial
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt.

# Não podemos assumir que lm.fit1 e lm.fit3 são significativamente melhores que
# lm.fitAnt devido a questões de overfitting. É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.567614
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(model1)
# Adjusted R-squared: 0.586564 -> Melhorou bastante
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1))
summary(models)
```

```
## O MAE apresenta-se mais baixo no model1.
## O RMSE apresenta-se mais baixo no model1.
## O Adjusted R-squared apresenta-se mais elevado no model1.
## Como as diferenças são consideráveis, o melhor modelo é então o model1:
## glyhb~stab.glu + age


#------------------------------------------------------------------------------


# 3 variáveis -> stab.glu + age + time.ppn
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn)
treino = treino[complete.cases(treino),]

lm.fit.Time.ppn = lm(glyhb~time.ppn, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Time.ppn)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Time.ppn) # Adjusted R-squared: -0.001485
# Sozinha parece não adicionar precisão ao modelo


lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5739

lm.fit1 <- lm(glyhb~stab.glu + age + time.ppn, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5786 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fit1

# Não podemos assumir que lm.fit1 é significativamente melhor que lm.fitAnt devido a
# questões de overfitting.
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.615430
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + age + time.ppn, data = treino, method = "lm", trControl = train.control)
print(model1)
# Adjusted R-squared: 0.615881 -> Melhorou muito pouco. Não vale a pena aumentar
# a complexidade
```

```r
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

# Melhor modelo:
models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1))
summary(models)
## O MAE apresenta-se mais baixo no modelAnt.
## O RMSE apresenta-se mais baixo no model1 mas a diferença é muito reduzida.
## O Adjusted R-squared apresenta-se mais elevado no model1 mas melhorou muito pouco.
## O melhor modelo é então o model5 -> glyhb~stab.glu + age



#-------------------------------------------------------------------------

# 3 variáveis -> stab.glu + age + hip
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,hip)
treino = treino[complete.cases(treino),]

lm.fit.hip = lm(glyhb~hip, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.hip)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal mas também não se
summary(lm.fit.hip) # Adjusted R-squared: 0.01746
# Sozinha parece não adicionar precisão ao modelo


lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5739

lm.fit1 <- lm(glyhb~stab.glu + age + hip, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5748 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt

## O melhor modelo é então o modelAnt -> glyhb~stab.glu + age


#-------------------------------------------------------------------------

# 3 Var -> stab.glu + age + bp.1s
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,bp.1s)
treino = treino[complete.cases(treino),]

lm.fit.bp.1s = lm(glyhb~bp.1s, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.bp.1s)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
```

```r
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal.
summary(lm.fit.bp.1s) # Adjusted R-squared: 0.03667
# Sozinha parece não adicionar precisão ao modelo



lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5739

lm.fit1 <- lm(glyhb~stab.glu + age + bp.1s, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5742 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt

## O melhor modelo é então o modelAnt -> glyhb~stab.glu + age


#------------------------------------------------------------------------

####################
# Melhor modelo!!! #
####################
# 2 variáveis -> glyhb~stab.glu + age
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn,hip,bp.1s)
treino = treino[complete.cases(treino),]

lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735

set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
melhorModelo <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(melhorModelo)
## Para este dataset (treino) -> Adjusted R-squared: 0.579255
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)


#-------------------------------------------------------------------------


############################
# Modelos com combinações #
############################

### REGSUBSETS
# 2 Var -> stab.glu + chol
# 3 Var -> stab.glu + ratio + age
# 4 Var -> stab.glu + ratio + age + time.ppn
# 5 Var -> stab.glu + ratio + age + time.ppn + chol

treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn,hip,bp.1s)
treino = treino[complete.cases(treino),]

# 2 Var -> stab.glu + age
```

```r
lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735
lm.fit1 <- lm(glyhb~stab.glu + age + stab.glu:age, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5792 -> melhorou muito pouco

## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.579255
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + age + stab.glu:age, data = treino, method = "lm", trControl = train.
    control)
print(modelo1)
# Adjusted R-squared: 0.571200 -> Pior
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

## Melhor modelo até agora: glyhb~stab.glu + age

#-------------------------------------------------------------------------

# 3 Var -> stab.glu + age + time.ppn
lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735

lm.fit1 <- lm(glyhb~stab.glu + age + time.ppn + stab.glu:time.ppn, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5935 -> melhorou bastante
lm.fit2 <- lm(glyhb~stab.glu + age + time.ppn + age:time.ppn, data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5834 -> melhorou um pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit2)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fitAnt


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.579255
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
    trControl = train.control)
print(modelo1)
# Adjusted R-squared: 0.590397 -> Melhorou um pouco
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)

```

```r
set.seed(1)
modelo2 <- train(glyhb~stab.glu + age + time.ppn + age:time.ppn, data = treino, method = "lm", trControl =
    train.control)
print(modelo2)
# Adjusted R-squared: 0.5789102 -> Piorou
1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)

## Não compensa adicionar 2 preditores para melhorar a previsão em apenas 1%
## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age


#-------------------------------------------------------------------------

# 4 Var -> stab.glu + age + time.ppn + hip
lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735
lm.fit1 <- lm(glyhb~stab.glu + age + hip + stab.glu:hip, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5736 -> igual
lm.fit2 <- lm(glyhb~stab.glu + age + hip + age:hip, data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5741 -> melhorou muito pouco
lm.fit3 <- lm(glyhb~stab.glu + age + time.ppn + hip + time.ppn:hip, data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5882 -> melhorou um pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit2)
# Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit3)
# Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fitAnt

# Não podemos assumir que lm.fit3 é significativamente
# melhor que lm.fitAnt devido a questões de overfitting.
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.579255
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + age + time.ppn + hip + time.ppn:hip, data = treino, method = "lm",
    trControl = train.control)
print(modelo1)
# Adjusted R-squared: 0.5910418 -> melhorou um pouco
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)


## Não compensa adicionar 2 preditores para melhorar a previsão em apenas 1%
## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age

```

```r
#-------------------------------------------------------------------------

# 5 Var -> stab.glu + age + time.ppn + hip + bp.1s
lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735
lm.fit1 <- lm(glyhb~stab.glu + age + bp.1s + stab.glu:bp.1s, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5719 -> Piorou
lm.fit2 <- lm(glyhb~stab.glu + age + bp.1s + age:bp.1s, data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5719 -> Piorou
lm.fit3 <- lm(glyhb~stab.glu + age + bp.1s + time.ppn + time.ppn:bp.1s, data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5789 -> Melhorou muito pouco
lm.fit4 <- lm(glyhb~stab.glu + age + bp.1s + hip + hip:bp.1s, data = treino)
summary(lm.fit4) # Adjusted R-squared: 0.5728 -> Piorou

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit2)
# Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit3)
# Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit4)
# Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
# que o modelo lm.fitAnt

# Não podemos assumir que lm.fit1 e lm.fit2 são significativamente
# melhores que lm.fitAnt devido a questões de overfitting.
# É necessário verificar por validação cruzada!


## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age

#-------------------------------------------------------------------------

####################################
# Melhor modelo com combinações!!! #
####################################

# Combinações não parecem favorecer o modelo. Deste modo o melhor modelo é:
lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5735

train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.579255
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

#-------------------------------------------------------------------------



## REGSUBSETS ##

```

```r
lm.fit <- lm(glyhb ~ chol + stab.glu + hdl + ratio + age + gender + height + weight + frame + waist + hip +
    time.ppn, data = nossoDiabetes, x = TRUE, y = TRUE)
summary(lm.fit)


reg = regsubsets(glyhb~.-diabetesB, data=nossoDiabetes, nvmax=10)
summary(reg)
# 1 Var -> stab.glu
# 2 Var -> stab.glu + chol
# 3 Var -> stab.glu + ratio + age
# 4 Var -> stab.glu + ratio + age + time.ppn
# 5 Var -> stab.glu + ratio + age + time.ppn + chol


## Teste com 1 var do regsubsets ## stab.glu
set.seed(1)
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu)
treino = treino[complete.cases(treino),]

lm.fit.Stab.glu = lm(glyhb~stab.glu, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Stab.glu)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos estão
# espalhados em volta de uma linha horizontal, o que indica que em princípio não
# existem relações não lineares
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos parecem s
# seguir uma distribuição normal

lm.fit1 <- lm.fit.Stab.glu
summary(lm.fit1) # Adjusted R-squared: 0.5602

## Elevar o grau de stab.glu (grau 2)
lm.fit2 <- lm(glyhb~poly(stab.glu, 2, raw=TRUE), data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5678 -> Melhorou um pouco

# F-test parcial
anova(lm.fit1, lm.fit2)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fit1.

## Elevar o grau de stab.glu (grau 3)
lm.fit3 <- lm(glyhb~poly(stab.glu, 3, raw=TRUE), data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5877 -> Melhorou bastante

# F-test parcial
anova(lm.fit1, lm.fit3)
# Rejeitamos também a hipotese nula de que o modelo lm.fit3 não é significativamente
# melhor que o modelo lm.fit1.

# Não podemos assumir que estes modelos são significativamente melhores que lm.fit1
# devido a questões de overfitting. É necessário verificar por validação cruzada!

## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
model1 <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
print(model1)
```

```
# Adjusted R-squared: 0.567815
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)

set.seed(1)
model2 <- train(glyhb~poly(stab.glu, 2, raw=TRUE), data = treino, method = "lm", trControl = train.control)
print(model2)
# Adjusted R-squared: 0.562181
1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
model3 <- train(glyhb~poly(stab.glu, 3, raw=TRUE), data = treino, method = "lm", trControl = train.control)
print(model3)
# Adjusted R-squared: 0.558148
1-(1-mean(model3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

models <- resamples(list("stab.glu(grau1)"=model1,
                         "stab.glu(grau2)"=model2,
                         "stab.glu(grau3)"=model3))
summary(models)
## O MAE apresenta-se mais baixo no model3.
## O RMSE apresenta-se mais baixo no model1.
## O Adjusted R-squared apresenta-se mais elevado no model1.
## O melhor modelo é então o model1 -> glyhb~stab.glu

#-------------------------------------------------------------------------------

# 2 variáveis -> stab.glu + chol
set.seed(1)
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,chol)
treino = treino[complete.cases(treino),]

lm.fit.Chol = lm(glyhb~chol, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Chol)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha "reta", o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Chol) # Adjusted R-squared: 0.05863
# Parece adicionar precisão ao modelo

lm.fitAnt <- lm(glyhb~stab.glu, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5602

lm.fit1 <- lm(glyhb~stab.glu + chol, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5759 -> Melhorou bastante

# F-test parcial
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt.

lm.fit2 <- lm(glyhb~stab.glu + poly(chol, 2, raw=TRUE), data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5759 -> Não melhorou

# F-test parcial
anova(lm.fitAnt, lm.fit2)
```

```r
# Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fit1.

lm.fit3 <- lm(glyhb~stab.glu + poly(chol, 3, raw=TRUE), data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5826 -> Melhorou muito pouco

# F-test parcial
anova(lm.fit1, lm.fit3)
# Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fit1.

# Não podemos assumir que lm.fit1 e lm.fit3 são significativamente melhores que
# lm.fitAnt devido a questões de overfitting. É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.567614
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + chol, data = treino, method = "lm", trControl = train.control)
print(model1)
# Adjusted R-squared: 0.581092 -> Melhor
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
model2 <- train(glyhb~stab.glu + poly(chol, 3, raw=TRUE), data = treino, method = "lm", trControl = train.
    control)
print(model2)
# Adjusted R-squared: 0.577321 -> Piorou (overfitting)
1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)

models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1,
                         "model2"=model2))
summary(models)
## O MAE apresenta-se mais baixo no model1.
## O RMSE apresenta-se mais baixo no model2 mas o model1 fica muito próximo.
## O Adjusted R-squared apresenta-se mais elevado no model1.
## O melhor modelo é então o model1.


#------------------------------------------------------------------------------


# 3 variáveis -> stab.glu + ratio + age
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,chol,ratio,age)
treino = treino[complete.cases(treino),]

lm.fit.Ratio = lm(glyhb~ratio, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Ratio)
```

```r
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha um pouco curvada, o que
# indica que poderão existir relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Ratio) # Adjusted R-squared: 0.1057
# Parece adicionar precisão ao modelo

lm.fit.Age = lm(glyhb~age, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Age)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Age) # Adjusted R-squared: 0.1128
# Parece adicionar precisão ao modelo

lm.fitAnt <- lm(glyhb~stab.glu + chol, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5759

lm.fit1 <- lm(glyhb~stab.glu + ratio, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5738 -> Pior

lm.fit2 <- lm(glyhb~stab.glu + poly(ratio, 2, raw=TRUE), data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5810 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fit1, lm.fit2)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fit1

lm.fit3 <- lm(glyhb~stab.glu + poly(ratio, 3, raw=TRUE), data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5823 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fit2, lm.fit3)
# Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fit2.

lm.fit4 <- lm(glyhb~stab.glu + age, data = treino)
summary(lm.fit4) # Adjusted R-squared: 0.5742 -> Pior

lm.fit5 <- lm(glyhb~stab.glu + poly(age, 2, raw=TRUE), data = treino)
summary(lm.fit5) # Adjusted R-squared: 0.5754 -> Pior

# F-test parcial:
anova(lm.fit4, lm.fit5)
# Aceitamos a hipotese nula de que o modelo lm.fit5 não é significativamente melhor
# que o modelo lm.fit4.

lm.fit6 <- lm(glyhb~stab.glu + poly(age, 3, raw=TRUE), data = treino)
summary(lm.fit6) # Adjusted R-squared: 0.5798 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fit4, lm.fit6)
```

```r
# Rejeitamos a hipotese nula de que o modelo lm.fit6 não é significativamente melhor
# que o modelo lm.fit4

# Não podemos assumir que lm.fit2 é significativamente melhor que lm.fit1 e que
# lm.fit6 é significativamente melhor que lm.fit4 e lm.fit6 devido a questões de
# overfitting. Também não podemos comparar os modelos lm.fit1 a lm.fit6 com o modelo
# lm.fitAnt pois ao passar de duas variáveis para três variáveis foi removida uma
# variável (chol) e adicionadas duas novas (ratio e age).
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu + chol, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.581092
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + ratio, data = treino, method = "lm", trControl = train.control)
print(model1)
# Adjusted R-squared: 0.578752 -> Pior
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
model2 <- train(glyhb~stab.glu + poly(ratio, 2, raw=TRUE), data = treino, method = "lm", trControl = train.
    control)
print(model2)
# Adjusted R-squared: 0.575364 -> Pior (overfitting)
1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

set.seed(1)
model3 <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
print(model3)
# Adjusted R-squared: 0.584609 -> Ligeiramente melhor
1-(1-mean(model3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)

set.seed(1)
model4 <- train(glyhb~stab.glu + poly(age, 2, raw=TRUE), data = treino, method = "lm", trControl = train.
    control)
print(model4)
# Adjusted R-squared: 0.586733 -> Melhorou muito pouco (nao vale a pena)
1-(1-mean(model4$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

## É melhor usar ratio e age lineares.
set.seed(1)
model5 <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(model5)
# Adjusted R-squared: 0.594697 -> Malhorou ligeiramente
1-(1-mean(model5$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

# Melhores modelos:
models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1,
                         "model3"=model3,
```

```
                        "model5"=model5))
summary(models)
## O MAE apresenta-se mais baixo no model5.
## O RMSE apresenta-se mais baixo no model5.
## O Adjusted R-squared apresenta-se mais elevado no model5.
## O melhor modelo é então o model5 -> glyhb~stab.glu + ratio + age



#------------------------------------------------------------------------

# 4 variáveis -> stab.glu + ratio + age + time.ppn
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn)
treino = treino[complete.cases(treino),]

lm.fit.Time.ppn = lm(glyhb~time.ppn, data = treino)
par(mfrow=c(2,2))
plot(lm.fit.Time.ppn)
## Explicar o plot aqui
# Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
# estão uniformemente espalhados em volta de uma linha reta, o que indica
# que em princípio não existem relações não lineares.
# Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
# seguir uma distribuição normal
summary(lm.fit.Time.ppn) # Adjusted R-squared: -0.001485
# Sozinha parece não adicionar precisão ao modelo


lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5855

lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5907 -> Melhorou ligeiramente

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt

lm.fit2 <- lm(glyhb~stab.glu + ratio + age + poly(time.ppn, 2, raw=TRUE), data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5897 -> Piorou

# F-test parcial:
anova(lm.fit1, lm.fit2)
# Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fit1.

lm.fit3 <- lm(glyhb~stab.glu + ratio + age + poly(time.ppn, 3, raw=TRUE), data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5915 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fit1, lm.fit3)
# Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fit1.

# Não podemos assumir que lm.fit1 é significativamente melhor que lm.fitAnt devido
# a questões de overfitting. É necessário verificar por validação cruzada!


```

```r
## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + ratio + age + time.ppn, data = treino, method = "lm", trControl = train.
    control)
print(model1)
# Adjusted R-squared: 0.626305 -> Melhorou muito pouco
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)


# Melhores modelos:
models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1))
summary(models)
## O MAE apresenta-se ligeiramente mais baixo no model1.
## O RMSE apresenta-se ligeiramente mais baixo no model1.
## O Adjusted R-squared apresenta-se ligeiramente mais elevado no model1.
## Não parece compensar aumentar a complexidade para ter resultados ligeiramente
## melhores
## O melhor modelo é então o modelAnt -> glyhb~stab.glu + ratio + age


#------------------------------------------------------------------------

# 5 variáveis -> stab.glu + ratio + age + time.ppn + chol
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
treino = treino[complete.cases(treino),]

## Análise do chol já foi feita em cima
summary(lm.fit.Chol) # Adjusted R-squared: 0.05863
# Parece adicionar precisão ao modelo


lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5855

lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5934 -> Melhorou ligeiramente

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt

lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 2, raw=TRUE), data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5934 -> Melhorou ligeiramente

# F-test parcial:
anova(lm.fit1, lm.fit2)
# Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
```

```
# que o modelo lm.fit1

lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 3, raw=TRUE), data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5984 -> Melhorou ligeiramente

# F-test parcial:
anova(lm.fit1, lm.fit3)
# Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fit1

# Não podemos assumir que lm.fit1 e lm.fit3 são significativamente melhores que
# lm.fitAnt devido a questões de overfitting.
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

## Melhor modelo anterior:
set.seed(1)
modelAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(modelAnt)
# Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

# Modelos a testar:
set.seed(1)
model1 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol, data = treino, method = "lm", trControl =
    train.control)
print(model1)
# Adjusted R-squared: 0.626159 -> Melhorou muito pouco
1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

set.seed(1)
model2 <- train(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 3, raw=TRUE), data = treino, method = "
    lm", trControl = train.control)
print(model2)
# Adjusted R-squared: 0.620654 -> Piorou
1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-6-1)

# Melhores modelos:
models <- resamples(list("modelAnt"=modelAnt,
                         "model1"=model1,
                         "model2"=model2))
summary(models)
## O MAE apresenta-se ligeiramente mais baixo no model1.
## O RMSE apresenta-se ligeiramente mais baixo no modelAnt.
## O Adjusted R-squared apresenta-se ligeiramente mais elevado no model1.
## Não parece compensar aumentar a complexidade para ter resultados ligeiramente
## melhores
## O melhor modelo é então o modelAnt -> glyhb~stab.glu + ratio + age


#------------------------------------------------------------------------

####################
# Melhor modelo!!! #
####################
# 3 variáveis -> glyhb~stab.glu + ratio + age
```

```r
treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
treino = treino[complete.cases(treino),]

lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5858

set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
melhorModelo <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(melhorModelo)
## Para este dataset (treino) -> Adjusted R-squared: 0.594697
# Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)


#-------------------------------------------------------------------------


############################
# Modelos com combinações #
############################

### REGSUBSETS
# 2 Var -> stab.glu + chol
# 3 Var -> stab.glu + ratio + age
# 4 Var -> stab.glu + ratio + age + time.ppn
# 5 Var -> stab.glu + ratio + age + time.ppn + chol

treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)

treino = treino[complete.cases(treino),]

# 2 Var -> stab.glu + chol
lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5858
lm.fit1 <- lm(glyhb~stab.glu + chol + stab.glu:chol, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.5811 -> baixou

## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.624422
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + chol + stab.glu:chol, data = treino, method = "lm", trControl = train.
    control)
print(modelo1)
# Adjusted R-squared: 0.612082 -> Pior
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

#-------------------------------------------------------------------------------

# 3 Var -> stab.glu + ratio + age
lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5858
```

```
957
958 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + stab.glu:ratio, data = treino)
959 summary(lm.fit1) # Adjusted R-squared: 0.5910 -> melhorou um pouco
960
961 # F-test parcial:
962 anova(lm.fitAnt, lm.fit1)
963 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
964 # que o modelo lm.fitAnt
965
966 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + stab.glu:age, data = treino)
967 summary(lm.fit2) # Adjusted R-squared: 0.5896 -> melhorou muito pouco
968
969 # F-test parcial:
970 anova(lm.fitAnt, lm.fit2)
971 # Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
972 # que o modelo lm.fitAnt
973
974 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + stab.glu:ratio + stab.glu:age, data = treino)
975 summary(lm.fit3) # Adjusted R-squared: 0.5959 -> melhorou um pouco
976
977 # F-test parcial:
978 anova(lm.fitAnt, lm.fit3)
979 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
980 # que o modelo lm.fitAnt
981
982 lm.fit4 <- lm(glyhb~stab.glu + ratio + age + ratio:age, data = treino)
983 summary(lm.fit4) # Adjusted R-squared: 0.5849 -> Pior
984
985 # F-test parcial:
986 anova(lm.fitAnt, lm.fit4)
987 # Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
988 # que o modelo lm.fitAnt
989
990 ## Verificar validação cruzada
991 train.control <- trainControl(method = "cv", number = 10)
992
993 set.seed(1)
994 modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
995 print(modeloAnt)
996 # Adjusted R-squared: 0.624422
997 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
998
999 set.seed(1)
1000 modelo1 <- train(glyhb~stab.glu + ratio + age + stab.glu:ratio, data = treino, method = "lm", trControl =
        train.control)
1001 print(modelo1)
1002 # Adjusted R-squared: 0.613438 -> Pior
1003 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
1004
1005 set.seed(1)
1006 modelo2 <- train(glyhb~stab.glu + ratio + age + stab.glu:age, data = treino, method = "lm", trControl =
        train.control)
1007 print(modelo2)
1008 # Adjusted R-squared: 0.614949 -> Pior
1009 1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
1010
1011 set.seed(1)
1012 modelo3 <- train(glyhb~stab.glu + ratio + age + ratio:age, data = treino, method = "lm", trControl = train.
        control)
```

```
print(modelo3)
# Adjusted R-squared: 0.621185 -> Pior
1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)

## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + ratio + age

#------------------------------------------------------------------------

# 4 Var -> stab.glu + ratio + age + time.ppn
lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.5858
lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
summary(lm.fit1) # Adjusted R-squared: 0.6036 -> melhorou um pouco
lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + ratio:time.ppn, data = treino)
summary(lm.fit2) # Adjusted R-squared: 0.5917 -> melhorou muito pouco
lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + age:time.ppn, data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.5973 -> melhorou um pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit2)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit3)
# Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fitAnt

# Não podemos assumir que lm.fit1, lm.fit2, lm.fit3 e lm.fit4 são significativamente
# melhores que lm.fitAnt devido a questões de overfitting.
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.624422
1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
    trControl = train.control)
print(modelo1)
# Adjusted R-squared: 0.637304 -> melhorou um pouco
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

set.seed(1)
modelo2 <- train(glyhb~stab.glu + ratio + age + time.ppn + ratio:time.ppn, data = treino, method = "lm",
    trControl = train.control)
print(modelo2)
# Adjusted R-squared: 0.623103 -> Pior
1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

set.seed(1)
modelo3 <- train(glyhb~stab.glu + ratio + age + time.ppn + age:time.ppn, data = treino, method = "lm",
```

```
    trControl = train.control)
print(modelo3)
# Adjusted R-squared: 0.625895 -> Melhorou muito pouco
1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

## O melhor modelo passa a ser o modelo1 -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn

#-------------------------------------------------------------------------

# 5 Var -> stab.glu + ratio + age + time.ppn + chol
lm.fitAnt <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
summary(lm.fitAnt) # Adjusted R-squared: 0.6036
lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + stab.glu:chol, data =
    treino)
summary(lm.fit1) # Adjusted R-squared: 0.6083 -> Melhorou muito pouco
lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + ratio:chol, data = treino
    )
summary(lm.fit2) # Adjusted R-squared: 0.6142 -> Melhorou um pouco
lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + age:chol, data = treino)
summary(lm.fit3) # Adjusted R-squared: 0.6044 -> Melhorou muito pouco
lm.fit4 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + time.ppn:chol, data =
    treino)
summary(lm.fit4) # Adjusted R-squared: 0.6042 -> Melhorou muito pouco

# F-test parcial:
anova(lm.fitAnt, lm.fit1)
# Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit2)
# Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit3)
# Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
# que o modelo lm.fitAnt
anova(lm.fitAnt, lm.fit4)
# Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
# que o modelo lm.fitAnt

# Não podemos assumir que lm.fit1 e lm.fit2 são significativamente
# melhores que lm.fitAnt devido a questões de overfitting.
# É necessário verificar por validação cruzada!


## Verificar validação cruzada
train.control <- trainControl(method = "cv", number = 10)

set.seed(1)
modeloAnt <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm"
    , trControl = train.control)
print(modeloAnt)
# Adjusted R-squared: 0.637304
1-(1-mean(modeloAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

set.seed(1)
modelo1 <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
    trControl = train.control)
print(modelo1)
# Adjusted R-squared: 0.637304 -> melhorou um pouco
1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
```

```
1123
1124  set.seed(1)
1125  modelo2 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + stab.glu:chol, data =
          treino, method = "lm", trControl = train.control)
1126  print(modelo2)
1127  # Adjusted R-squared: 0.635052 -> Pior (overfitting)
1128  1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-7-1)
1129
1130  set.seed(1)
1131  modelo3 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + ratio:chol, method = "
          lm", trControl = train.control)
1132  print(modelo3)
1133  # Adjusted R-squared: 0.623915 -> Pior (overfitting)
1134  1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-7-1)
1135
1136  ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn
1137
1138  #-------------------------------------------------------------------------------
1139
1140  ####################################
1141  # Melhor modelo com combinações!!! #
1142  ####################################
1143  # 5 variáveis -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn
1144  treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
1145  treino = treino[complete.cases(treino),]
1146
1147  lm.fitAnt <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
1148  summary(lm.fitAnt) # Adjusted R-squared: 0.6036
1149
1150  set.seed(1)
1151  melhorModelo <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "
          lm", trControl = train.control)
1152  print(melhorModelo)
1153  # Adjusted R-squared: 0.637304
1154  1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1155
1156  #-------------------------------------------------------------------------------
1157
1158  # Contudo é necessário adicionar duas novas variáveis (time.ppn e stab.glu:time.ppn)
1159  # para subir o Adjusted R-squared de 0.624422 para 0.637304
1160
1161  set.seed(1)
1162  modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
1163  print(modeloAnt)
1164  # Adjusted R-squared: 0.624422
1165  1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
1166
1167  ## VS
1168
1169  set.seed(1)
1170  melhorModelo <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "
          lm", trControl = train.control)
1171  print(melhorModelo)
1172  # Adjusted R-squared: 0.637304
1173  1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1174
1175  ## Nós achamos que não compensa aumentar essas duas variáveis para ter um ganho tão
1176  # pouco significativo pelo que determinados o melhor modelo como sendo:
1177  # glyhb ~ stab.glu + ratio + age
```

Regressão Logística

```
1
2
3
4  ### CROSS VALIDATION GLM
5
6  ## Retirar os nulls
7  dados = dplyr::select(diabetes, diabetesB,bp.1s,stab.glu,hdl,chol,age,ratio,stab.glu,weight,height,waist,hip
       ,gender,location,frame)
8  dados2 = dados[complete.cases(dados),]
9  attach(dados2)
10
11 ## Utilizando a nossa analise de variaveis as mais importantes são: STAB.GLU age bp.1s ratio, waist
12 dados2 = nossoDiabetes
13 ## Melhor erro ponderaro: 3.46 , th:0.4
14 set.seed(1)
15 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu,dados2)
16 glm.fit=glm(diabetesB~stab.glu,data=dados,family=binomial)
17 plot(glm.fit)
18
19 melhor = melhorIndice(thr)
20
21 ## Melhor erro ponderaro: 3.43 , th:0.4
22 set.seed(1)
23 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age,dados2)
24 melhor = melhorIndice(thr)
25
26 ## 1p - Melhor erro ponderaro: 3.54 , th:0.4
27 set.seed(1)
28 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+age+stab.glu:age,dados2)
29 melhor = melhorIndice(thr)
30 thr[40,]
31
32 ## Melhor erro ponderaro: 3.52 , th:0.2
33 set.seed(1)
34 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s,dados2)
35 melhor = melhorIndice(thr)
36
37 ## Melhor erro ponderaro: 3.44 , th:0.4
38 set.seed(1)
39 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio,dados2)
40 melhor = melhorIndice(thr)
41
42 ## Melhor erro ponderaro: 3.47 , th:0.4
43 set.seed(1)
44 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio+waist,dados2)
45 melhor = melhorIndice(thr)
46
47 ## Melhor erro ponderaro: 3.47 , th:0.4
48 set.seed(1)
49 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio+waist+weight,dados2)
50 melhor = melhorIndice(thr)
51
52 ##############################################################################################
53 #Utilizar o regsubsets ########################################################################
54
55 regfit.best=regsubsets(diabetesB~.,data=dados2,nvmax=19)
56 summary(regfit.best)
```

```
57
58
59
60  ## 1p - Melhor erro ponderaro: 3.46 , th:0.4
61  set.seed(1)
62  thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~poly(stab.glu,3),dados2)
63  melhor = melhorIndice(thr)
64
65  ## 1p - Melhor erro ponderaro: 3.57 , th:0.4
66  set.seed(1)
67  thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2),dados2)
68  melhor = melhorIndice(thr)
69  thr[40,]
70
71
72
73
74
75  ## 1p - Melhor erro ponderaro: 3.48 , th:0.4
76  set.seed(1)
77  thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
78  melhor = melhorIndice(thr)
79
80
81  ## 1p - Melhor erro ponderaro: 3.55 , th:0.4
82  set.seed(1)
83  thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
84  melhor = melhorIndice(thr)
85
86  ## 1p - Melhor erro ponderaro: 3.55 , th:0.4 anterior com 0.0.1 th
87  set.seed(1)
88  thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
89  melhor = melhorIndice(thr)
90
91  ## 1p - Melhor erro ponderaro: 3.57 , th:0.4 Stab.glu + ratio
92  set.seed(1)
93  thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2)+ratio,dados2)
94  melhor = melhorIndice(thr)
95  thr[40,]
96
97  ## 1p - Melhor erro ponderaro: 3.54 , th:0.4 Stab.glu + ratio
98  set.seed(1)
99  thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2)+ratio+age,nossoDiabetes)
100 melhor = melhorIndice(thr)
101 thr[40,]
102 thr[21,]
103
104
105 ## 2p -Melhor erro ponderaro: 3.43 , th:0.2
106 set.seed(1)
107 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ age +stab.glu:age ,dados2)
108 melhor = melhorIndice(thr)
109
110
111 ## 2p -Melhor erro ponderaro: 3.48 , th:0.2
112 set.seed(1)
113 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio,dados2)
114 melhor = melhorIndice(thr)
115
```

```
## 2p -Melhor erro ponderaro: 3.55 , th:0.2
set.seed(1)
thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio,dados2)
melhor = melhorIndice(thr)

set.seed(1)
thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio,dados2)
melhor = melhorIndice(thr)

## 3p - Melhor erro ponderaro: 3.46 , th:0.4
set.seed(1)
thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age,dados2)
melhor = melhorIndice(thr)

## 4p - Melhor erro ponderaro: 3.46 , th:0.4
set.seed(1)
thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age+gender,dados2)
melhor = melhorIndice(thr)

## 5p - Melhor erro ponderaro: 3.47 , th:0.4
set.seed(1)
thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age+gender+waist,dados2)
melhor = melhorIndice(thr)
```

KNN

```
## NOSSA ANALISE ##
## Teste com 1 var ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu)
treino = treino[complete.cases(treino),]

resultadoKnnCV1 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK1 = melhorIndice(resultadoKnnCV1)

## Teste com 2 vars ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age)
treino = treino[complete.cases(treino),]

resultadoKnnCV2 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK2 = melhorIndice(resultadoKnnCV2)

## Teste com 3 vars ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s)
treino = treino[complete.cases(treino),]

resultadoKnnCV3 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK3 = melhorIndice(resultadoKnnCV3)

## Teste com 4 vars ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s, ratio)
treino = treino[complete.cases(treino),]

resultadoKnnCV4 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK4 = melhorIndice(resultadoKnnCV4)

```

```
## Teste com 5 vars ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s, ratio, waist)
treino = treino[complete.cases(treino),]

resultadoKnnCV5 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK5 = melhorIndice(resultadoKnnCV5)

## o melhor é com 3 e k=10 ##

## ordena todos para selecionar os 5 melhores ##
valores = rep(1,dim(resultadoKnnCV1)[1])
res1 = data.frame(resultadoKnnCV1,valores)
valores = rep(2,dim(resultadoKnnCV1)[1])
res2 = data.frame(resultadoKnnCV2,valores)
valores = rep(3,dim(resultadoKnnCV1)[1])
res3 = data.frame(resultadoKnnCV3,valores)
valores = rep(4,dim(resultadoKnnCV1)[1])
res4 = data.frame(resultadoKnnCV4,valores)
valores = rep(5,dim(resultadoKnnCV1)[1])
res5 = data.frame(resultadoKnnCV5,valores)

todos = rbind(res1,res2,res3,res4,res5)
todosC = consomeValores(todos)
todosOrd = todosC[order(-todosC$valor),]



## REGSUBSETS ##

## Teste com 1 var do regsubsets ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu)
treino = treino[complete.cases(treino),]

resultadoKnnCV1 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK1 = melhorIndice(resultadoKnnCV1)

## Teste com 2 vars do regsubsets ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol)
treino = treino[complete.cases(treino),]

resultadoKnnCV2 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK2 = melhorIndice(resultadoKnnCV2)

## Teste com 3 vars do regsubsets ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio, age)
treino = treino[complete.cases(treino),]

resultadoKnnCV3 = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK3 = melhorIndice(resultadoKnnCV3)

## Teste com 4 vars do regsubsets ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio, age, time.ppn)
treino = treino[complete.cases(treino),]

resultadoKnnCV4 = geraKNNCV(treino,treino$diabetesB,1,20)
```

```

melhorK4 = melhorIndice(resultadoKnnCV4)

## com o metodo forward a 3 e 4 dá diferente ##
## com 3 ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol, age)
treino = treino[complete.cases(treino),]

resultadoKnnCV3F = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK3F = melhorIndice(resultadoKnnCV3F)

## com 4 ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol, age, time.ppn)
treino = treino[complete.cases(treino),]

resultadoKnnCV4F = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK4F = melhorIndice(resultadoKnnCV4F)

## com o metodo backward a segunda dá diferente ##
treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio)
treino = treino[complete.cases(treino),]

resultadoKnnCV2B = geraKNNCV(treino,treino$diabetesB,1,20)

melhorK2B = melhorIndice(resultadoKnnCV2B)

## ordena todos para selecionar os 5 melhores ##
valores = rep(1,dim(resultadoKnnCV1)[1])
res1 = data.frame(resultadoKnnCV1,valores)
valores = rep(2,dim(resultadoKnnCV1)[1])
res2 = data.frame(resultadoKnnCV2,valores)
valores = rep(3,dim(resultadoKnnCV1)[1])
res3 = data.frame(resultadoKnnCV3,valores)
valores = rep(4,dim(resultadoKnnCV1)[1])
res4 = data.frame(resultadoKnnCV4,valores)
valores = rep(5,dim(resultadoKnnCV1)[1])
res3F = data.frame(resultadoKnnCV3F,valores)
valores = rep(6,dim(resultadoKnnCV1)[1])
res4F = data.frame(resultadoKnnCV4F,valores)
valores = rep(7,dim(resultadoKnnCV1)[1])
res2B = data.frame(resultadoKnnCV2B,valores)

todos = rbind(res1,res2,res3,res4,res3F,res4F,res2B)
todosC = consomeValores(todos)
todosOrd = todosC[order(-todosC$valor),]
## o melhor é com k=9 e 2 vars, no entanto não são melhores ##
## que o nossa pelo que não será usado ##
```

QDA e LDA

```

## QDA
treino = dplyr::select(nossoDiabetes, diabetesB, age)
treino = treino[complete.cases(treino),]

qda1 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age,treino)
melhorQ1 = melhorIndice(qda1)
```

```
## acerto muito baixo em torno dos 71%

treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s)
treino = treino[complete.cases(treino),]

qda2 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age+bp.1s,treino)
melhorQ2 = melhorIndice(qda2)
qda2[12,]
## acerto muito baixo em torno dos 71%

treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s, waist)
treino = treino[complete.cases(treino),]

qda3 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age+bp.1s+waist,treino)
melhorQ3 = melhorIndice(qda3)
## acerto muito baixo em torno dos 72,25%


#### LDA
treino = dplyr::select(nossoDiabetes, diabetesB, age)
treino = treino[complete.cases(treino),]

lda1 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age,treino)
melhorL1 = melhorIndice(lda1)
## acerto na ordem dos 71,75%

treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s)
treino = treino[complete.cases(treino),]

lda2 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age+bp.1s,treino)
melhorL2 = melhorIndice(lda2)
## acerto muito baixo em torno dos 71,75%

treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s, waist)
treino = treino[complete.cases(treino),]

lda3 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age+bp.1s+waist,treino)
melhorL3 = melhorIndice(lda3)
## acerto muito baixo em torno dos 72,25%
```

Seleção do melhor modelo

```
## Dividir em teste e treino ##

set.seed(1)

teste=sample(dim(nossoDiabetes)[1],dim(nossoDiabetes)[1]/4)
diabetesTeste = nossoDiabetes[teste,]
diabetesTreino = nossoDiabetes[-teste,]
summary(diabetesTeste)
summary(diabetesTreino)



## Selecionar os melhores modelos ##
## knn ##

treinoA = dplyr::select(diabetesTreino, diabetesB,stab.glu, age, bp.1s)
summary(treinoA)
```

```
treinoA = treinoA[complete.cases(treinoA),]
summary(treinoA)
testeA = dplyr::select(diabetesTeste, diabetesB,stab.glu, age, bp.1s)
summary(testeA)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

set.seed(1)
knnK10 = knn(treinoA[,-1],testeA[,-1],treinoA$diabetesB,k=10)
table(knnK10,testeA$diabetesB)
## 3.5991 ##
set.seed(1)
knnK12 = knn(treinoA[,-1],testeA[,-1],treinoA$diabetesB,k=12)
table(knnK12,testeA$diabetesB)
## 3.4744 ##

## glm ##
glm.fit1 = glm(diabetesB~poly(stab.glu,2)+ratio,data=diabetesTreino,family=binomial)
summary(glm.fit1)
plot(glm.fit1)

testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, ratio)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

glm.pred1 = predict(glm.fit1,testeA,type="response")
glm.pred1 = ifelse(glm.pred1 > 0.39, 1, 0)

table(glm.pred1,testeA$diabetesB)
## 3.5991 ##

## Ratio n é importante, p-valor muito baixo ##
diabetesTreino[149,] ## é o 195
auxTreino = diabetesTreino#[-149,]
auxTreino[149,]
glm.fit1 = glm(diabetesB~poly(stab.glu,2),data=diabetesTreino,family=binomial)
summary(glm.fit1)
plot(glm.fit1)

testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

glm.pred1 = predict(glm.fit1,testeA,type="response")
glm.pred1 = ifelse(glm.pred1 > 0.39, 1, 0)

table(glm.pred1,testeA$diabetesB)
## 3.5991 ##

## lm ##
diabetesTreino[149,] ## é o 195
auxTreino = diabetesTreino#[-149,]
auxTreino[149,]

lm.fitAnt <- lm(glyhb~stab.glu + age, data = auxTreino)
summary(lm.fitAnt)
plot(lm.fitAnt)


```

```
testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, age)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

lm.pred1 = predict(lm.fitAnt,testeA,interval="prediction")

valoresLM1 = transformaBinaria(lm.pred1[,1])

table(valoresLM1,testeA$diabetesB)

## 3.6318 ##

lm.fit2 <- lm(glyhb~stab.glu + ratio + age, data = diabetesTreino)
summary(lm.fit2)
plot(lm.fit2)

testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, ratio, age)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

lm.pred2 = predict(lm.fit2,testeA,interval="prediction")

valoresLM2 = transformaBinaria(lm.pred2[,1])

table(valoresLM2,testeA$diabetesB)

## 3.6318 ##
```

Resultados e respostas

```
lm.fitAnt <- lm(glyhb~stab.glu + age, data = auxTreino)
summary(lm.fitAnt)
plot(lm.fitAnt)


testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, age)
testeA = testeA[complete.cases(testeA),]
summary(testeA)

lm.pred1 = predict(lm.fitAnt,testeA,interval="prediction")

valoresLM1 = transformaBinaria(lm.pred1[,1])

table(valoresLM1,testeA$diabetesB)

## Resposta ##
summary(nossoDiabetes$stab.glu)


##meus dados o ano passado JOSE PEDRO
stab.glu = 91
age = 20
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .89999)
## com ~90% de confiança posso afirmar que não tenho diabetes

##
stab.glu = 91
age = 15
```

```
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .911)
## com 91.1% de confiança posso afirmar que não tenho diabetes

##
stab.glu = 91
age = 50
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .785)
## com 50 anos a confinaça desce para ~78.5% com a mesma stab.glu

##
stab.glu = 91
age = 70
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .666)
## com 70 anos a confinaça desce para ~66.6% com a mesma stab.glu

##
stab.glu = 61
age = 20
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .971)
## com a mesma idade e menos stab.glu a confiança aumentou para 97.1

##
stab.glu = 121
age = 20
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .721)
## com a mesma idade e mais stab.glu a confiança desceu para 72.1

##
stab.glu = 151
age = 20
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .40)
## com a mesma idade e mais stab.glu a confiança desceu para 40

## ter diabetes ##
##
stab.glu = 191
age = 60
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .547)
## com estes valores temos 54.7% de confiança que tem diabetes

##
stab.glu = 191
age = 40
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .37)
## com estes valores temos 37% de confiança que tem diabetes

##
stab.glu = 191
age = 20
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .167)
```

```
## com 20 anos a confianca baixa para 16.7%

##
stab.glu = 191
age = 75
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .66)
## aumentando a idade para 75 a confiança sobe para 66%


##
stab.glu = 151
age = 60
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .005)
## diminuindo a stab.glu para 151 a confiança passa para 0.5%


##
stab.glu = 171
age = 60
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .299)
## diminuindo a stab.glu para 171 a confiança passa para 29.9%

##
stab.glu = 211
age = 60
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .738)
## aumentando a stab.glu para 211 a confiança passa para 73.8%

##
stab.glu = 231
age = 60
testeResposta = data.frame(stab.glu,age)
lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .863)
## aumentando a stab.glu para 231 a confiança passa para 86.3%


## Incidencia pessoas com mais e menos 50 anos ##
menos50 = nossoDiabetes[nossoDiabetes$age<50,]
mais50 = nossoDiabetes[nossoDiabetes$age>=50,]

quantosMenos0 = length(menos50$diabetesB[!is.na(menos50$diabetesB) & menos50$diabetesB==0])
quantosMenos1 = length(menos50$diabetesB[!is.na(menos50$diabetesB) & menos50$diabetesB==1])

quantosMenos0/(quantosMenos0+quantosMenos1) ##94,22% dos q tem menos de 50 n tem diabetes

quantosMais0 = length(menos50$diabetesB[!is.na(mais50$diabetesB) & mais50$diabetesB==0])
quantosMais1 = length(menos50$diabetesB[!is.na(mais50$diabetesB) & mais50$diabetesB==1])

quantosMais0/(quantosMais0+quantosMais1) ##70,40% dos q tem mais de 50 n tem diabetes

quantosMais0/(quantosMais0+quantosMenos0) ## 42.55% dos q n tem diabetes tem mais de 50 anos

quantosMais1/(quantosMais1+quantosMenos1) ## 83.54% dos q tem diabetes tem mais de 50 anos

## Incidencia location ##
```

```
148 cidadeTeste = dplyr::select(nossoDiabetes,diabetesB,location)
149 cidadeTeste = cidadeTeste[complete.cases(cidadeTeste),]
150
151 barplot(table(cidadeTeste$diabetesB,cidadeTeste$location),col=c( "cornflowerblue" , "cornsilk3" ),beside=T,
        legend.text=c("Nao␣ter","Ter"))
152 table(cidadeTeste$diabetesB,cidadeTeste$location)
153 31/190 ## em buckingham 16,32% tem diabetes
154 29/200 ## em louisa 14.5% tem diabetes
155
156 31/60 ##dos q tem diabetes 51.67% pertencem a buckingham
157 159/(159+171) ##dos q n tem diabetes 48.18%pertencem a buckingham
158 ## as incidencias sao bastante parecidas ##
159
160
161 ## Resultado melhor com lm ##
```

Definição de funções auxiliares

```
 1 graficoThreshold <- function(inferior,superior,delta, funcao, formula, dados){
 2   num = ((superior - inferior) / delta) + 2
 3   aux = 0
 4   auxT = 0
 5   comeco = inferior
 6   v = rep(0,num)
 7   acerto = rep(0,num)
 8   acertoP = rep(0,num)
 9   acertoN = rep(0,num)
10
11   for(a in 1: num){
12     comeco = inferior + ((a-1)*delta)
13     resultados = funcao(dados,formula,comeco)
14     v[a] = comeco
15     acerto[a] = resultados[1,1]
16     acertoP[a] = resultados[1,2]
17     acertoN[a] = resultados[1,3]
18   }
19   estrutura <- data.frame(v,acerto,acertoP,acertoN)
20 }
21
22 melhorThreshold <- function(inferior,superior,delta, fit, teste){
23   num = ((superior - inferior) / delta) + 2
24   acerto = 0
25   valor = -1
26   comeco = inferior
27
28   for(a in 1: num){
29     comeco = inferior + ((a-1)*delta)
30     resultados = rep(NA,183)
31     resultados[fit<=comeco] = 0
32     resultados[fit>comeco] = 1
33     condicaoT <- resultados == teste
34
35     mediaAuxT = mean(teste==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
36     ##algo diferente tendo em conta aquilo q disse na analise
37     if(mediaAuxT >= 0.65){
38       mediaAuxP = length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/length(resultados[!is.na
             (teste) & teste == 0])
39
40       if(mediaAuxP >= 0.65){
```

```
41          mediaAuxN = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/length(resultados[!is.
                na(teste) & teste == 1])
42
43          if(mediaAuxN >= 0.65){
44            mediaAux = (2*mediaAuxT) + mediaAuxP + mediaAuxN ##aqui pode-se
45            ##truncar os valores para x casas decimais
46
47            if(mediaAux > acerto){
48              acerto = mediaAux
49              valor = comeco
50            }
51
52          }
53
54        }
55      }
56
57    }
58    resultados = rep(NA,183)
59    resultados[fit<=valor] = 0
60    resultados[fit>valor] = 1
61    condicaoT <- resultados == teste
62
63    acertoT = mean(teste==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
64
65    acertoN = length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/length(resultados[!is.na(teste
          ) & teste == 0])
66
67    acertoP = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/length(resultados[!is.na(teste
          ) & teste == 1])
68
69    acerto = acerto / 4
70    estrutura <- data.frame(valor, acertoT, acertoN, acertoP,acerto)
71  }
72
73  geraKNNCV <- function(treinoX, treinoY, inicio, fim){
74    num = fim - inicio + 1
75    v = rep(0,num)
76    t = rep(0,num)
77    fp = rep(0,num)
78    fn = rep(0,num)
79    for(a in 1:num){
80      i = inicio + a - 1
81      set.seed(1)
82      resultados = knn.cv(treino[,-1],treinoY,k=a)
83
84      v[a] = i
85      v[a] = i
86      mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
87      ##algo diferente tendo em conta aquilo q disse na analise
88      t[a] = mediaAux
89
90      condicaoT <- resultados == treinoY
91      divide = length(resultados[!is.na(treinoY) & treinoY == 0])
92      if(divide != 0){
93        mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
94        fn[a] = mediaAux
95      }
96      divide = length(resultados[!is.na(treinoY) & treinoY == 1])
```

```
97
98      if(divide!=0){
99        mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
100       fp[a] = mediaAux
101     }
102
103   }
104   estrutura <- data.frame(v,t,fp,fn)
105 }
106
107 melhorIndice = function(dados){
108   data = dados
109   indice = -1
110   valor = 0
111   for(i in 1:dim(data)[1]){
112     valorAux = (2*data[i,2]) + data[i,3] + data[i,4]
113
114     if(valorAux > valor){
115       valor = valorAux
116       indice = i
117     }
118   }
119
120   resultado = data.frame(indice,valor)
121
122 }
123
124 glmCV = function(dados,formula,thr){
125   treino = dados[complete.cases(dados),]
126
127   quantos = dim(treino)[1]/10
128   acerto = rep(NA,quantos)
129   acertoP = rep(NA,quantos)
130   acertoN = rep(NA,quantos)
131
132   for(a in 1:quantos){
133     teste = sample(dim(treino)[1],10)
134     dadosTeste = treino[teste, ]
135     treino = treino[-teste, ]
136
137     if(a == 1){
138
139       glm.fit = glm(formula, data=treino, family=binomial)
140       glm.probs=predict(glm.fit, newdata=dadosTeste, type="response")
141       resultados = ifelse(glm.probs > thr, 1, 0)
142       treinoY = dadosTeste[,1]
143
144       mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
145       ##algo diferente tendo em conta aquilo q disse na analise
146       acerto[a] = mediaAux
147
148       condicaoT <- resultados == treinoY
149       divide = length(resultados[!is.na(treinoY) & treinoY == 0])
150       if(divide != 0){
151         mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
152         acertoN[a] = mediaAux
153       }
154       divide = length(resultados[!is.na(treinoY) & treinoY == 1])
155
```

```
      if(divide!=0){
        mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
        acertoP[a] = mediaAux
      }
      jaUsados = dadosTeste
    }
    else{
      paraTreinar = rbind(treino,jaUsados)
      glm.fit = glm(formula, data=paraTreinar, family=binomial)
      glm.probs=predict(glm.fit, newdata=dadosTeste, type="response")
      resultados = ifelse(glm.probs > thr, 1, 0)
      treinoY = dadosTeste[,1]

      mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
      ##algo diferente tendo em conta aquilo q disse na analise
      acerto[a] = mediaAux

      condicaoT <- resultados == treinoY
      divide = length(resultados[!is.na(treinoY) & treinoY == 0])
      if(divide != 0){
        mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide

        acertoN[a] = mediaAux
      }
      divide = length(resultados[!is.na(treinoY) & treinoY == 1])
      if(divide != 0){
        mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide

        acertoP[a] = mediaAux
      }
      jaUsados = rbind(jaUsados,dadosTeste)
    }
  }
  acertoMedio = mean(acerto, na.rm = TRUE)
  acertoMedioP = mean(acertoP, na.rm = TRUE)
  acertoMedioN = mean(acertoN, na.rm = TRUE)

  data.frame(acertoMedio,acertoMedioP,acertoMedioN)
}

qdaCV = function(dados,fitFrom,thr){
  treino = dados[complete.cases(dados),]

  quantos = dim(treino)[1]/10
  acerto = rep(NA,quantos)
  acertoP = rep(NA,quantos)
  acertoN = rep(NA,quantos)

  for(a in 1:quantos){
    set.seed(1)
    teste = sample(dim(treino)[1],10)
    dadosTeste = treino[teste, ]
    treino = treino[-teste, ]

    if(a == 1){
      set.seed(1)
      qda.fit=qda(fitFrom,data=treino)
      qda.pred=predict(qda.fit,dadosTeste)
      resultadosAux = qda.pred$posterior[,2]
```

```
      resultados = ifelse(resultadosAux >= thr, 1, 0)
      treinoY = dadosTeste[,1]

      mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
      ##algo diferente tendo em conta aquilo q disse na analise
      acerto[a] = mediaAux

      condicaoT <- resultados == treinoY
      divide = length(resultados[!is.na(treinoY) & treinoY == 0])
      if(divide != 0){
        mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
        acertoN[a] = mediaAux
      }
      divide = length(resultados[!is.na(treinoY) & treinoY == 1])

      if(divide!=0){
        mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
        acertoP[a] = mediaAux
      }
      jaUsados = dadosTeste
    }
    else{
      paraTreinar = rbind(treino,jaUsados)
      qda.fit=qda(fitFrom,data=paraTreinar)
      qda.pred=predict(qda.fit,dadosTeste)
      resultadosAux = qda.pred$posterior[,2]
      resultados = ifelse(resultadosAux >= thr, 1, 0)
      treinoY = dadosTeste[,1]

      mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
      ##algo diferente tendo em conta aquilo q disse na analise
      acerto[a] = mediaAux

      condicaoT <- resultados == treinoY
      divide = length(resultados[!is.na(treinoY) & treinoY == 0])
      if(divide != 0){
        mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide

        acertoN[a] = mediaAux
      }
      divide = length(resultados[!is.na(treinoY) & treinoY == 1])
      if(divide != 0){
        mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide

        acertoP[a] = mediaAux
      }
      jaUsados = rbind(jaUsados,dadosTeste)
    }
  }
  acertoMedio = mean(acerto, na.rm = TRUE)
  acertoMedioP = mean(acertoP, na.rm = TRUE)
  acertoMedioN = mean(acertoN, na.rm = TRUE)

  data.frame(acertoMedio,acertoMedioP,acertoMedioN)
}

ldaCV = function(dados,fitFrom,thr){
  treino = dados[complete.cases(dados),]

```

```
quantos = dim(treino)[1]/10
acerto = rep(NA,quantos)
acertoP = rep(NA,quantos)
acertoN = rep(NA,quantos)

for(a in 1:quantos){
  set.seed(1)
  teste = sample(dim(treino)[1],10)
  dadosTeste = treino[teste, ]
  treino = treino[-teste, ]

  if(a == 1){
    set.seed(1)
    qda.fit=lda(fitFrom,data=treino)
    qda.pred=predict(qda.fit,dadosTeste)
    resultadosAux = qda.pred$posterior[,2]
    resultados = ifelse(resultadosAux >= thr, 1, 0)
    treinoY = dadosTeste[,1]

    mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
    ##algo diferente tendo em conta aquilo q disse na analise
    acerto[a] = mediaAux

    condicaoT <- resultados == treinoY
    divide = length(resultados[!is.na(treinoY) & treinoY == 0])
    if(divide != 0){
      mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
      acertoN[a] = mediaAux
    }
    divide = length(resultados[!is.na(treinoY) & treinoY == 1])

    if(divide!=0){
      mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
      acertoP[a] = mediaAux
    }
    jaUsados = dadosTeste
  }
  else{
    paraTreinar = rbind(treino,jaUsados)
    qda.fit=lda(fitFrom,data=paraTreinar)
    qda.pred=predict(qda.fit,dadosTeste)
    resultadosAux = qda.pred$posterior[,2]
    resultados = ifelse(resultadosAux >= thr, 1, 0)
    treinoY = dadosTeste[,1]

    mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
    ##algo diferente tendo em conta aquilo q disse na analise
    acerto[a] = mediaAux

    condicaoT <- resultados == treinoY
    divide = length(resultados[!is.na(treinoY) & treinoY == 0])
    if(divide != 0){
      mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide

      acertoN[a] = mediaAux
    }
    divide = length(resultados[!is.na(treinoY) & treinoY == 1])
    if(divide != 0){
      mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
```

```
333
334         acertoP[a] = mediaAux
335       }
336       jaUsados = rbind(jaUsados,dadosTeste)
337     }
338   }
339   acertoMedio = mean(acerto, na.rm = TRUE)
340   acertoMedioP = mean(acertoP, na.rm = TRUE)
341   acertoMedioN = mean(acertoN, na.rm = TRUE)
342
343   data.frame(acertoMedio,acertoMedioP,acertoMedioN)
344 }
345
346 normaliza = function(dados){
347   data = dados
348   for(i in 1:dim(data)[2]){
349     if(class(data[,i])!="factor"){
350       data[,i] <- (data[,i] - min(data[,i], na.rm = TRUE)) / (max(data[,i],na.rm = TRUE)-min(data[,i],na.rm
            = TRUE))
351     }
352   }
353
354   resultado = data
355 }
356
357 consomeValores = function(dados){
358   indice = rep(NA,dim(dados)[1])
359   k = rep(NA,dim(dados)[1])
360   valor = rep(NA,dim(dados)[1])
361   for(i in 1:dim(dados)[1]){
362     indice[i] = dados[i,5]
363     k[i] = dados[i,1]
364     valor[i] = (2*dados[i,2]) + dados[i,3] + dados[i,4]
365   }
366
367   data.frame(indice,k,valor)
368 }
369
370 transformaBinaria = function(dados){
371   binaria <- rep(NA,length(dados))
372   binaria[dados > 7] = 1
373   binaria[dados <= 7] = 0
374   binaria
375 }
```

AA1

Apresentação do trabalho prático

14 de Dezembro de 2018

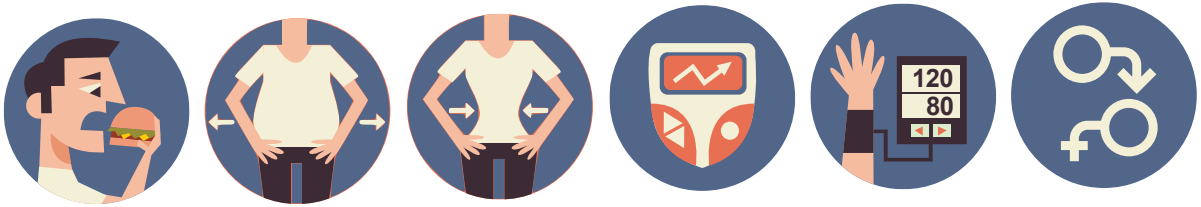
Dataset

Informações do Dataset:

- ✓ Nome: Diabetes
- ✓ Origem: EUA
- ✓ Data: 1997
- ✓ Quantidade: 403 obs. e 19 var

Variáveis

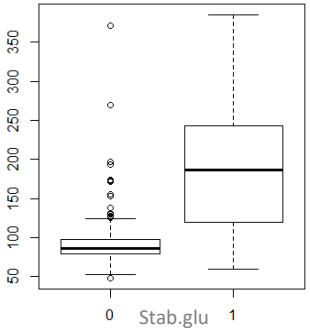
- ✓ Stab.glu
- ✓ Chol
- ✓ Time.ppn
- ✓ Hip
- ✓ Weight
- ✓ Height
- ✓ Bp.1s
- ✓ Bp.1d
- ✓ Bp.2s
- ✓ Bp.2d
- ✓ Gly.hb
- ✓ Ratio
- ✓ Frame
- ✓ Location
- ✓ Hdl
- ✓ Gender
- ✓ Waist

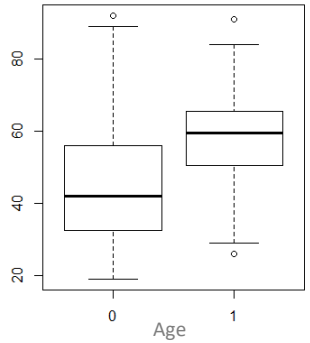


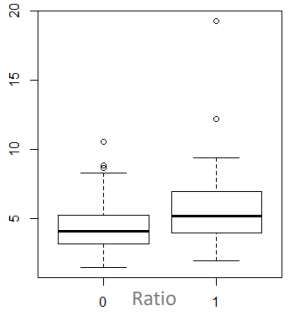
Análise exploratória

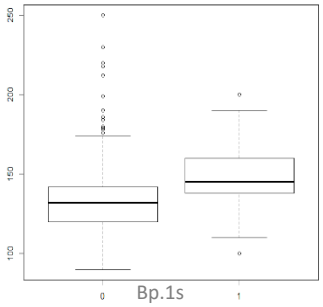
Análise de influencia

Var Binária

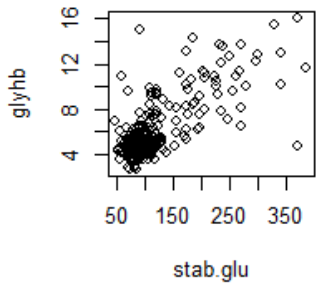


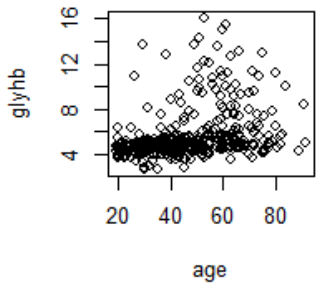


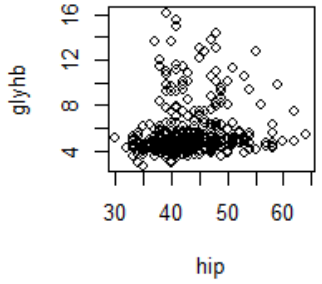


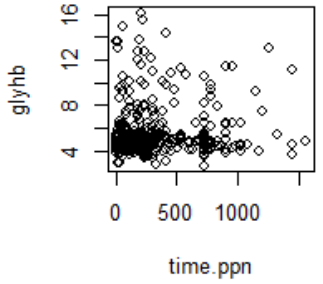


Gly.hb









Ranking variáveis

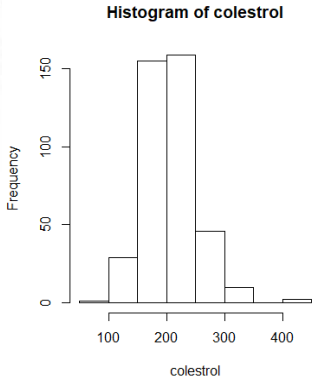
✓ Var Binária

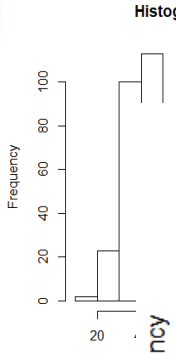
1. Stab.glu
2. Age
3. Bp.1s
4. Racio
5. Waist
6. Weight
7. Hip
8. Hdl
9. Chol
10. Bp.1d

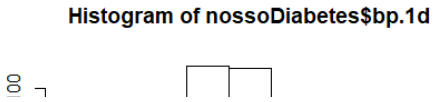
✓ Gly.hb

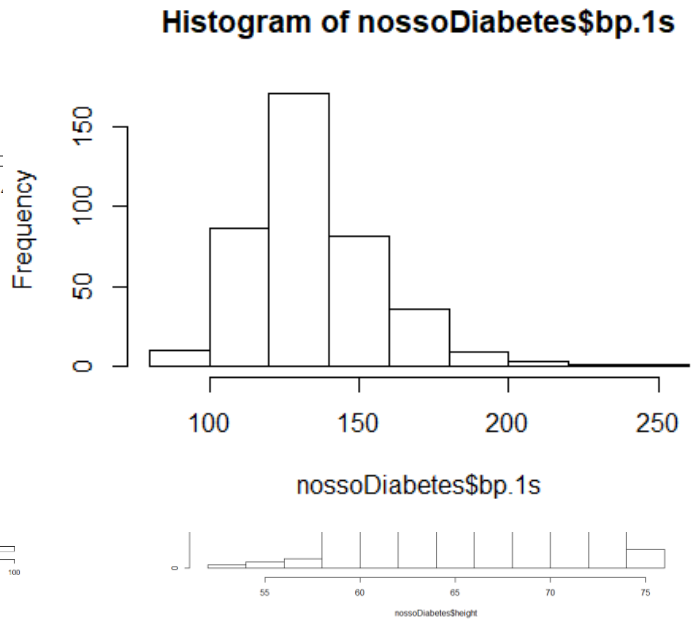
1. Stab.glu
2. Age
3. Time.ppn
4. Hip
5. Bp.1s
6. Waist
7. Bp.1d
8. Chol
9. Height
10. Weigth

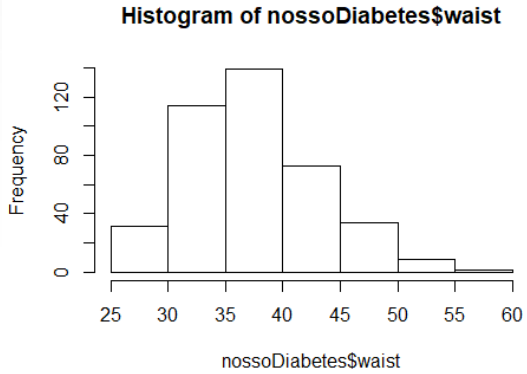
Análise de normalidade

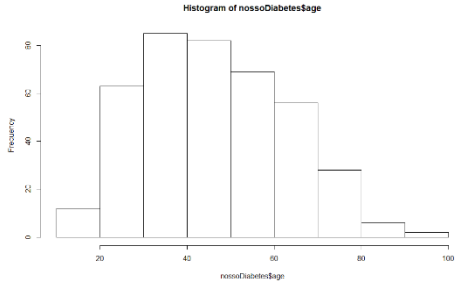


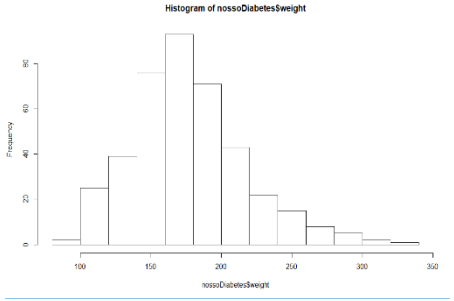












Seleção do modelo

LM vs GLM vs QDA vs KNN vs LDA

Regressão linear (LM)

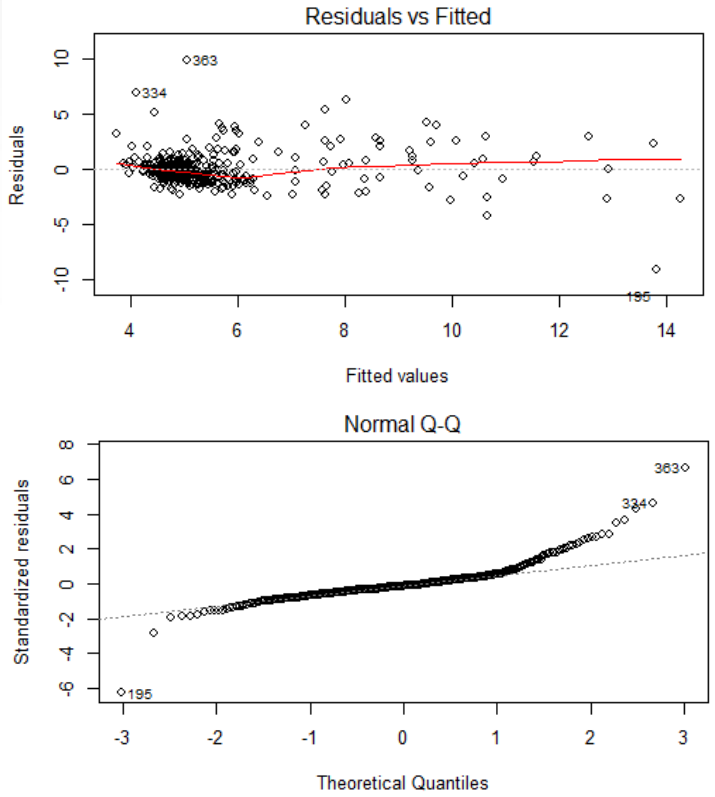
Modelos:

Dataset:

- stab.glu
  Adj. R-squared: 0.5602
- stab.glu^2
  Adj. R-squared: 0.5678
- stab.glu^3
  Adj. R-squared: 0.5877

CV(K=10):

- stab.glu
  Adj. R-squared: 0.5678
- stab.glu^2
  Adj. R-squared: 0.5621
- stab.glu^3
  Adj. R-squared: 0.5581

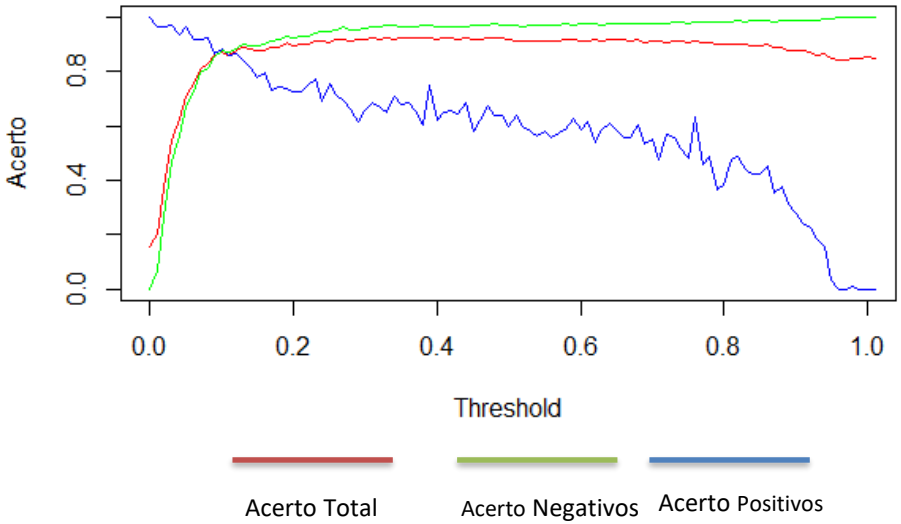


Regressão logística (GLM)

Modelos:

Fórmula	Thr	Acerto
Stab.glu^2	0.39	3.5714
Stab.glu^2+ratio	0.39	3.5714
Stab.glu+ratio + stab.glu x ratio	0.39	3.5520

Gráfico Threshold:



KNN

Nossa abordagem

Variáveis	K	Acerto ponderado
Stab.glu	14	3.4825
Stab.glu+age	9	3.4707
Stab.glu+age+bp.1s	10	3.5320
Stab.glu+age+bp.1s+ratio	12	3.5316
Stab.glu+age+bp.1s+ratio+waist	10	3.5307

Regsubsets

Variáveis	K	Acerto ponderado
Stab.glu+chol	9	3.4902
Stab.glu+ratio+age	16	3.4766
Stab.glu+ratio+age+time.ppn	3	3.4286
(F)stab.glu+chol+age	8	3.4684
(F)stab.glu+chol+age+time.ppn	4	3.4014
(B)stab.glu+ratio	15	3.4820

QDA E LDA

QDA

Variáveis	Threshold	Acerto Ponderado
Age	0.22	2.8265
Age+bp.1s	0.24	2.8260
Age+bp.1s+ waist	0.18	2.9115

LDA

Variáveis	Threshold	Acerto Ponderado
Age	0.18	2.8578
Age+bp.1s	0.26	2.8039
Age+bp.1s+ waist	0.16	2.9364

Seleção melhor modelo (CV)

Modelo	Regressão linear	Regressão logística	KNN
Fórmula	Stab.glu + age	stab.glu^2	stab.glu + age + bp.1s
Threshold	>7	0.39	K=10
Acerto ponderado	$adj.r^2 = 56.23\%$	**3.5714**	3.5320

Seleção melhor modelo
(Teste e Treino)

Modelo	Regressão linear	Regressão logística	KNN
Fórmula	Stab.glu + age	stab.glu^2	stab.glu + age + bp.1s
Threshold	>7	0.39	K=10
Acerto ponderado	**3.6318**	3.5991	3.5665

Modelo escolhido

Modelo: Regressão linear

- ✓ **Fórmula: glyhb = 1.693 + 0.027*stab.glu + 0.020*age**
- ✓ **Adjusted R^2 : 56.23%**
- ✓ **Acerto ponderado: 3.63**
- ✓ **Acerto total: 95.88%**
- ✓ **Acerto positivos: 71.43%**
- ✓ **Acerto negativos: 100%**

Discussão de resultados

- **Será que o colesterol/pressão arterial/tempo após refeição/fatores corporais afetam os diabetes?**
- **Qual fator corporal explica melhor o valor da diabete?**
- **Quais fatores influenciam mais o resultado final?**
- **De que forma os fatores selecionados para a explicação dos resultados o influenciam? (crescentemente, decrescentemente, linearmente)**

Discussão de resultados

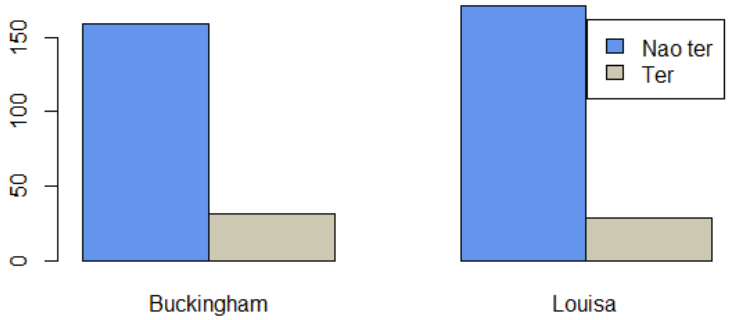
? **Qual a probabilidade (ou confiança no resultado) de uma pessoa com as caraterísticas X (por exemplo colesterol=180, altura=175, peso=67, etc.) ter diabetes?**

Stab.glu	Age	Confiança
151	60	0.5%
171	60	29.9%
191	60	54.7%
211	60	73.8%
231	60	86.3%

Stab.glu	Age	Confiança
191	20	16.7%
191	40	37%
191	60	54.7%
191	75	66%

Discussão de resultados

- **Qual a taxa de incidência em pessoas com menos e com mais de 50 anos?**
- **Qual a cidade apresenta maior incidência? (visto serem só dois podemos comparar)**
- **O resultado é mais exato utilizando um modelo de classificação ou de regressão (e de seguida classificando)?**



Incidência em pessoas com mais/menos do que 50 anos

- ✓ 94.22% das pessoas que têm menos de 50 anos não têm diabetes
- ✓ 70,4% das pessoas que têm mais de 50 anos não têm diabetes
- ✓ 42.55% dos que não têm diabetes têm mais de 50 anos
- ✓ 83.54% dos que têm diabetes têm mais de 50 anos

Q & A