Banco de dados - Body Performance

Contextualização

São dados que fornecem informações de medição para cada item dos dados nacionais de medição de condicionamento físico gerenciados pela National Sports Promotion Corporation para a comemoração olímpica de Seul.

O intuito da análise

O intuito desse relatório é apresentar o tema de Regressão Linear múltipla, sendo um modelo de análise utilizado quando modelamos a relação linear entre uma variável resposta e múltiplas variáveis preditoras.

Pacotes utilizados

```
library(tidyverse)
library(psych)
library(caTools)
library(corrplot)
```

O **Pacote Tidyverse** É uma coleção de pacotes R projetados para ciência de dados. Todos os pacotes compartilham uma filosofia de design, gramática e estruturas de dados subjacentes.Para mais informações `[Saiba Mais em](https://www.tidyverse.org/)` .

O **Pacote psych** Funções desse pacote são principalmente para análise multivariada e construção de escala usando análise fatorial, análise de componentes principais, análise de cluster e análise de confiabilidade, embora outras forneçam estatísticas descritivas básicas. `[Saiba Mais em](https://cran.r-project.org/web/packages/psych/index.html)` .

O **Pacote caTools** Contém várias funções utilitárias básicas, incluindo: funções estatísticas de janela móvel (rolando, em execução), leitura/gravação para arquivos binários GIF e ENVI, cálculo rápido de AUC, classificador LogitBoost, codificador/decodificador base64, soma e cumsum sem erros de arredondamento, etc. `[Saiba Mais em](https://https://www.rdocumentation.org/packages/corrplot/versions/0.92/)`

O **Pacote corrplot** é utilizado para visualizar uma matriz de correlação. Para mais informações `[Saiba mais em] (https://www.rdocumentation.org/packages/corrplot/versions/0.2-0/topics/corrplot)` .

Carregando a base de dados

```
df <- read.csv('bodyPerformance.csv',header = TRUE,sep = ',');head(df, 5)
```

	a…	gender	height_cm	weight_kg	body.fat_.	diastolic	systolic	gripForce
	<dbl>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	27	M	172.3	75.24	21.3	80	130	54.9
2	25	M	165.0	55.80	15.7	77	126	36.4
3	31	M	179.6	78.00	20.1	92	152	44.8
4	32	M	174.5	71.10	18.4	76	147	41.4
5	28	M	173.8	67.70	17.1	70	127	43.5

5 rows | 1-9 of 13 columns

Variáveis

Variável	Descrição
age	Idade do indivíduo
gender	Gênero do indivíduo
height_cm	Altura do indivíduo
weight_kg	Peso do indivíduo
body.fat_.	Percentual de gordura do corpo
weight_kg	Peso do indivíduo
diastolic	Pressão arterial diastólica é o menor valor encontrado durante a medida de pressão arterial.
Sístole	É o período de contração muscular das câmaras cardíacas que alterna com o período de repouso, diástole.
gripforce	Grip force é a força aplicada pela mão para puxar ou suspender objetos e é uma parte específica da força da mão.
sit and bend forward_cm	Sentar e dobrar para frente em centímetros
sit-ups counts	Contagens de abdominais
broad jump_cm	salto largo em centímetros
Class	Classe (ou Categoria)

Verificando o formato das variáveis

```
str(df)
```

```
## 'data.frame':    13393 obs. of  12 variables:
##  $ age                     : num  27 25 31 32 28 36 42 33 54 28 ...
##  $ gender                  : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 1 2 2 2 ...
##  $ height_cm               : num  172 165 180 174 174 ...
##  $ weight_kg               : num  75.2 55.8 78 71.1 67.7 ...
##  $ body.fat_.              : num  21.3 15.7 20.1 18.4 17.1 22 32.2 36.9 27.6 14.4 ...
##  $ diastolic               : num  80 77 92 76 70 64 72 84 85 81 ...
##  $ systolic                : num  130 126 152 147 127 119 135 137 165 156 ...
##  $ gripForce               : num  54.9 36.4 44.8 41.4 43.5 23.8 22.7 45.9 40.4 57.9 ...
##  $ sit.and.bend.forward_cm: num  18.4 16.3 12 15.2 27.1 21 0.8 12.3 18.6 12.1 ...
##  $ sit.ups.counts          : num  60 53 49 53 45 27 18 42 34 55 ...
##  $ broad.jump_cm           : num  217 229 181 219 217 153 146 234 148 213 ...
##  $ class                   : Factor w/ 4 levels "A","B","C","D": 3 1 3 2 2 2 4 2 3 2 ...
```

> Através do comando **str**, conseguimos visualizar o formato das variáveis do conjunto de dados **'bodyPerformance.csv'**. Sendo 10 variáveis numéricas e 2 como fator.

Verificando se o conjunto de dados apresenta valores ausentes

```
colSums(is.na(df))
```

```
##                   age                gender             height_cm
##                     0                     0                     0
##             weight_kg            body.fat_.              diastolic
##                     0                     0                     0
##              systolic             gripForce sit.and.bend.forward_cm
##                     0                     0                     0
##         sit.ups.counts         broad.jump_cm                 class
##                     0                     0                     0
```

Elaborando um modelo de Regressão Linear Multipla

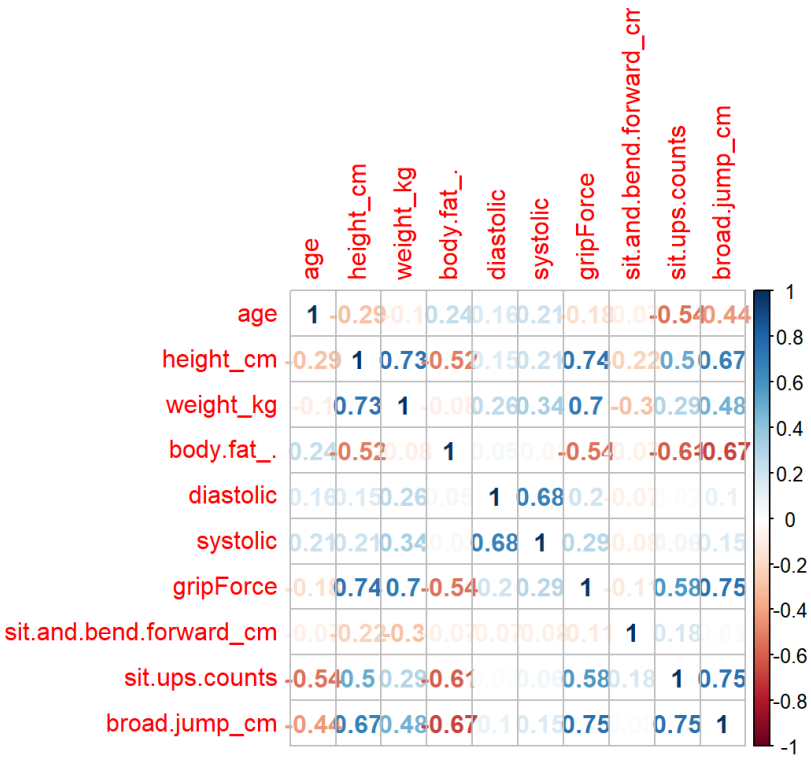
Vamos elaborar um modelo de Regressão linear múltipla para prever o valor da variável **gripForce** (ou força de aperto) com base no conjunto de variáveis contidos no banco de dados **bodyPerformance.csv**. Sendo assim o objetivo é estabelecer uma fórmula matemática entre à variável dependente (Y) e as variáveis independeres (x1,x2,x3…xn).

$$\mathrm{y} = \beta_0 + \beta_1(\mathrm{x1}) + \beta_2(\mathrm{x2}) + \beta_n(\mathrm{n}) + \epsilon$$

Variável	Descrição
y	O valor previsto da variável dependente
B0	O intercepto no eixo y (O valor de y quando todos os outros parâmetros são definidos como 0
b1	O coeficiente de regressão (B1 e B2) das primeiras variáveis independentes (também conhecido como o efeito que o aumento do valor da variável independente tem no valor de y previsto)
bn	O coeficiente de regressão da última variável independente
e	Erro do modelo (é a estimativa de variação que existe na estimação da variável dependente (ou y)

Matriz de Correlação

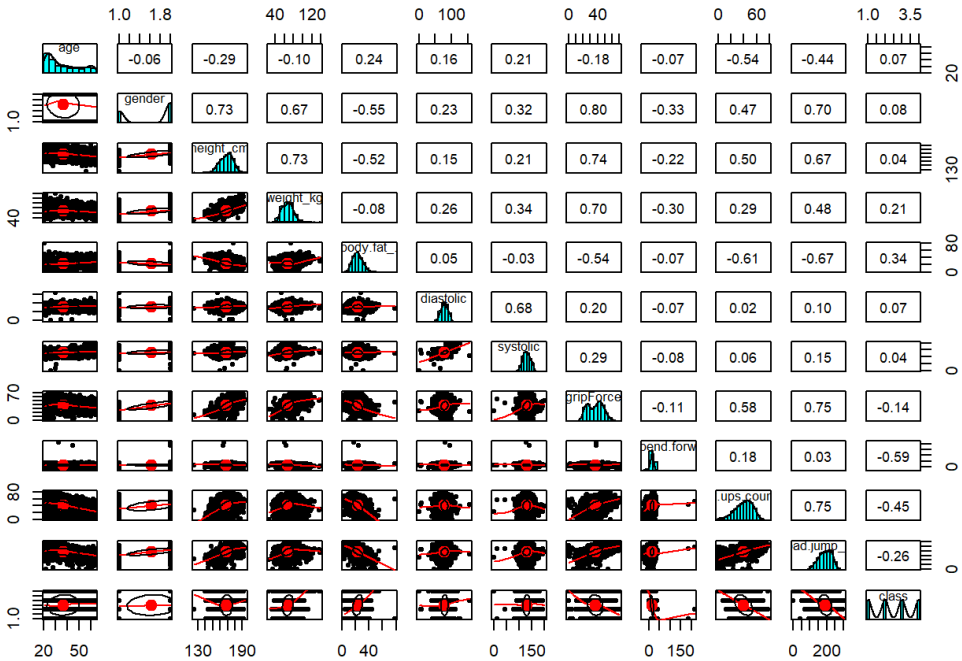
```
df.corr <- df[,-c(2,12)]

df.corr = cor(df.corr)

corrplot(df.corr, method="number")
```



Através da matriz de correlação conseguimos visualizar as relações entre variáveis. Sendo assim cada célula da tabela mostra a conexão entre os dois fatores. Vale ressaltar que uma correlação forte entre duas variáveis analisadas não implica haver uma relação de causa e efeito.

Visualizando as distribuições das variáveis do conjunto de dados

```
pairs.panels(df, col = "red")
```



Dividindo dados em treino e teste

```
split <- sample.split(df, SplitRatio = 0.7)

train_data <-  subset(df, split == TRUE)
test_data <- subset(df, split == FALSE)
```

Para a elaboração do modelo, vamos separar os dados em conjuntos de treino e teste. Sendo assim os dados de treino serão utilizados ao modelo para o treinamento e criação do modelo, neste caso selecionamos 70% do conjunto de dados para treinamento. Já o conjunto de dados teste serão introduzidos no modelo, apos sua criação, dessa forma simulando previsões reais que o modelo realizará, permitindo assim que o desempenho real seja verificado. Neste caso foram utilizados o restante do conjunto de dados para testar o modelo (ou 30%) do conjunto de dados.

Elaborando o modelo

```
model_1 <- lm(gripForce ~.,data = train_data );summary(model_1)
```

```
##
## Call:
## lm(formula = gripForce ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.680  -2.889  -0.126   2.891  20.797
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.397296   2.137323   1.590   0.1120
## age                       0.004055   0.005701   0.711   0.4769
## genderM                   6.653386   0.245465  27.105  < 2e-16 ***
## height_cm                -0.001229   0.012854  -0.096   0.9238
## weight_kg                 0.377438   0.009309  40.547  < 2e-16 ***
## body.fat_.               -0.249575   0.013656 -18.276  < 2e-16 ***
## diastolic                 0.027835   0.006601   4.217  2.5e-05 ***
## systolic                  0.003125   0.005045   0.620   0.5356
## sit.and.bend.forward_cm   0.013510   0.008115   1.665   0.0960 .
## sit.ups.counts            0.017055   0.007015   2.431   0.0151 *
## broad.jump_cm             0.042668   0.002778  15.359  < 2e-16 ***
## classB                   -1.383475   0.155287  -8.909  < 2e-16 ***
## classC                   -2.181487   0.170400 -12.802  < 2e-16 ***
## classD                   -3.837810   0.225942 -16.986  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.916 on 8914 degrees of freedom
## Multiple R-squared:  0.7864, Adjusted R-squared:  0.7861
## F-statistic:  2524 on 13 and 8914 DF,  p-value: < 2.2e-16
```

Quando utilizamos o comando **summay(model_1)**, visualizamos que as variáveis que não foram significantes para ajudar a predizer a variável dependente (ou Y), foram systolic,sit.and.bend.forward_cm e height_cm.

Sendo assim elaboraremos um segundo modelo sem essas variáveis listados no parágrafo anterior, de modo a verificar se com a remoção dessas variáveis conseguimos obter um modelo que melhor explica o R2.

Elaborando segundo modelo

```
model_2 <- lm(gripForce ~. -systolic -sit.and.bend.forward_cm - height_cm,data = train_data);summary(model_2)
```

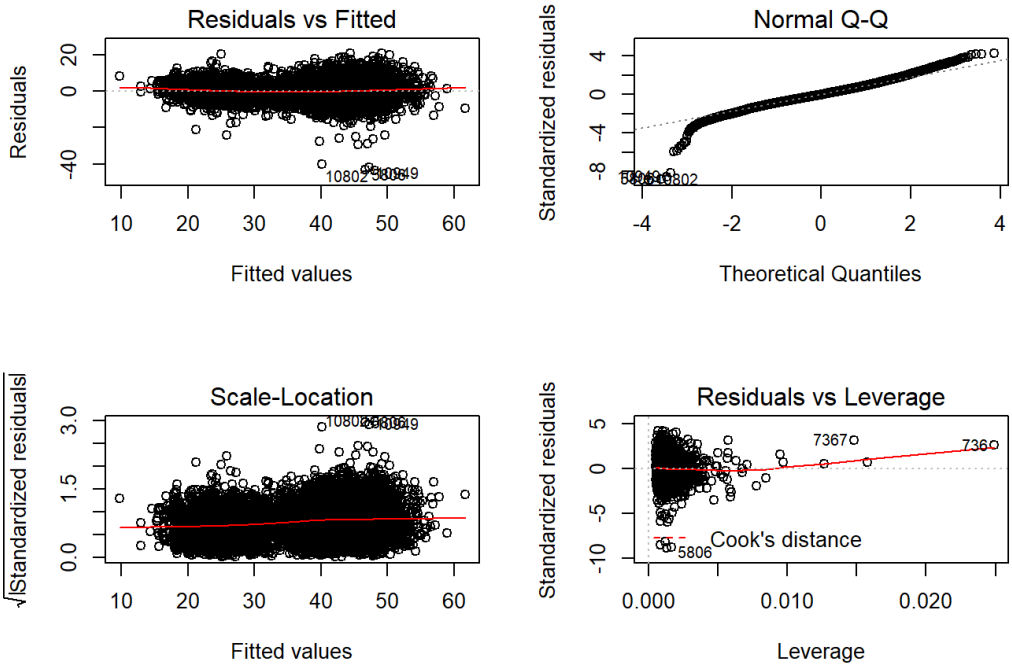
```
##
## Call:
## lm(formula = gripForce ~ . - systolic - sit.and.bend.forward_cm -
##     height_cm, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.667  -2.897  -0.143   2.908  20.868
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.511205   0.778833   4.508 6.62e-06 ***
## age             0.005040   0.005503   0.916   0.3597
## genderM         6.548034   0.234650  27.905  < 2e-16 ***
## weight_kg       0.377662   0.007145  52.858  < 2e-16 ***
## body.fat_.     -0.248943   0.012254 -20.315  < 2e-16 ***
## diastolic       0.030866   0.005192   5.944 2.88e-09 ***
## sit.ups.counts  0.017344   0.006987   2.482   0.0131 *
## broad.jump_cm   0.043260   0.002752  15.719  < 2e-16 ***
## classB         -1.423070   0.153541  -9.268  < 2e-16 ***
## classC         -2.259236   0.164484 -13.735  < 2e-16 ***
## classD         -4.001940   0.204865 -19.535  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.917 on 8917 degrees of freedom
## Multiple R-squared:  0.7863, Adjusted R-squared:  0.7861
## F-statistic:  3281 on 10 and 8917 DF,  p-value: < 2.2e-16
```

Explicando os termos apresentado na tabela acima.

Quando elaborado o segundo modelo, contendo somente as variáveis que foram significativas no modelo anterior, verificamos que todas as variáveis presentes na tabela demostram significância. Já quando analisamo o Multiple R-squared, visualizamos que o valor apresentou uma diferença mínima em relação ao modelo anterior.

1. **Estimate** : A coluna Estimate nos dá os coeficientes para cada variável independente no modelo de regressão.
2. **Std. Error** : Exibe o erro padrão da estimativa. Os números contidos nessa coluna mostra quanta variação existe em torno das estimativas do coeficiente do modelo (ou regressão).
3. **t value** : A estatística de teste usada na regressão linear é o valor t de um teste t bilateral . Quanto maior a estatística do teste, menos provável é que os resultados tenham ocorrido por acaso.
4. **Pr(>|t|)** : Essa coluna apresenta o valor p (ou probabilidade) de valor t calculado ter ocorrido por acaso se a hipótese nula de nenhum efeito do parâmetro fosse verdadeira.Como esses valores são tão baixos (p < 0,001 em ambos os casos), podemos rejeitar a hipótese nula e concluir que tanto ir de bicicleta para o trabalho quanto fumar provavelmente influenciam as taxas de doença cardíaca.
5. **Adjusted R-squared** : É uma medida estatística de quão próximos os dados estão da linha de regressão ajustada. Sendo assim é a porcentagem da variação da variável dependente (ou resposta) explicada pelo modelo linear. O R-squaed fica entre 0 e 100%.

```
par(mfrow = c(2,2))
plot(model_2)
```



Os gráficos de resíduos são usados para procurar padrões subjacentes nos resíduos que podem significar que o modelo tem um problema.

1. **Residuals vs Fitted** : Indica se existem padrões não lineares. Sendo assim quando elaboramos uma regressão linear correta, os dados precisam ser lineares, dessa forma visualizamos através do gráfico se essa condição foi atendida. Como podemos visualizar em nosso gráfico, verificamos que os resíduos ficam concentrados em torno da linha vermelha, não apresentando nenhum formato em curva ou (formato em V).
2. **Normal Q-Q** : Utilizamos o gráfico de normal Q-Q plot para visualizar se os resíduos são normalmente distribuídos. Em nosso caso os resíduos seguem perto de uma linha reta. Já quando visualizamos resíduos próximo da escala 4, podemos notar que os redisudos apresentam uma curva, isso indica que nosso modelo pode não apresentar bons desempenhos para valores mais altos.
3. **Scale-Location** : Utilizamos o gráfico para verificar a homoscedasticidade. Então basicamente, verifica se os resíduos têm variância igual ao da linha de regressão.
4. **Residuals vs Leverange** : Esse gráfico é utilizado para verificação de casos influentes no banco de dados. Um caso influente é aquele que, se removido afetara o modelo, de modo que a exclusão ou inclusão deve ser considerada. Entao basicamente o objetivo desse gráfico é identificar dados que tenha alta influência no modelo desenvolvido.

Utilizando a função predict no conjunto de dados teste

Intervalo de confiança

O intervalo de confiança reflete a incerteza em torno das previsões médias. Vale ressaltar que o intervalo de confiança na tabela abaixo é de 95%.

```
head(predict(model_2,newdata = test_data,interval = "confidence"))
```

```
##         fit      lwr      upr
## 1  43.94619 43.66162 44.23077
## 3  43.92968 43.60089 44.25848
## 9  36.30929 35.95969 36.65888
## 12 19.12222 18.64500 19.59944
## 13 21.75136 21.41277 22.08995
## 15 47.33741 47.05111 47.62371
```

1. **fit** Os valores de **Grip_Force** previstos para o conjunto de dados teste
2. **lwr** limite inferior do intervalo de confiança para os valores esperados
3. **upr** limite superior do intervalo de confiança para os valores esperados

Intervalo de previsão

O intervalo de previsão dá incerteza em torno de um único valor. Da mesma forma que os intervalos de confiança, os intervalos de predição podem ser calculados utilizando o comando abaixo.

```
head(predict(model_2, newdata = test_data,interval = "prediction"))
```

```
##         fit       lwr      upr
## 1  43.94619 34.304460 53.58793
## 3  43.92968 34.286541 53.57282
## 9  36.30929 26.665415 45.95316
## 12 19.12222  9.472875 28.77156
## 13 21.75136 12.107882 31.39484
## 15 47.33741 37.695621 56.97919
```

Referências

Regressão linear : (https://www.scribbr.com/statistics/linear-regression-in-r/) ;
(https://www.sheffield.ac.uk/polopoly_fs/1.536483!/file/MASH_multiple_regression_R.pdf)

Análise dos resíduos : (https://rpubs.com/iabrady/residual-analysis/)

Previsões do modelo :
(http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/)

Base de dados : (https://www.kaggle.com/kukuroo3/body-performance-data)

Contato

Caso o leitor tenha encontrado algum erro ou queira sugerir alguma mudança, ou sugestão entre em contato através do E-mail : charles.b.ribeiro@gmail.com (mailto:charles.b.ribeiro@gmail.com)

"Não tenha medo de cometer erros, tenha medo de não aprender com eles - Peter Jones