Análise Exploratória de Dados

André Teixeira
DEI
ISEP
Porto, Portugal
1190384@isep.ipp.pt

Ivo Oliveira
DEI
ISEP
Porto, Portugal
1190679@isep.ipp.pt

Palavras-chave—Análise, monitorização, precisão, rendimento, viaturas

I. Introduction

Este artigo científico foi desenvolvido no âmbito da disciplina de Análise de Dados em Informática (ANADI), da licenciatura de Engenharia Informática do Instituto Superior de Engenharia do Porto (ISEP). O projeto foi desenvolvido ao longo das aulas, o tema sendo a Análise Exploratória de Dados. O trabalho foi desenvolvido em R, usando a ferramenta R-Studio, que ajudou na análise, resolução e obtenção de resultados.

O relatório está divido em vários capítulos, cada um a focar num exercício proposto exceto o primeiro, da Introdução, e o último, da Conclusão.

II. Exercicio 1

A. *Alínea a)*

Primeiramente importamos os dados com a função *read.csv*.

```
DADOS1 <- read_csv("Dados excel/DADOS1.csv", skip =
2)
```

Inicialmente a coluna "Tempo_seg" dos "DADOS1" é convertida para o tipo numérico e atribuída à variável "segundos" permitindo trabalhar com os valores de tempo de forma numérica. Para acrescentar uma coluna aos dados com o tempo no sistema ***POSIXct*** recorremos ao uso da função *as.POSIXct*. onde "segundos " contém os valores de tempo em segundos, o fuso horário como GMT e a data específica de origem a partir da qual os valores de tempo são calculados.

```
segundos <- as.numeric(DADOS1$Tempo_seg)

DADOS1$Tempo <- as.POSIXct(segundos, tz = "GMT",
origin = "1970-01-01")
```

Por último, se quisermos visualizar os dados com a nova coluna com o tempo

```
View(DADOS1)
```

B. *Alínea b)*

Incialmente é criada uma nova variável chamada "data_selecionada" que contém apenas as linhas de "DADOS1" em que a data em formato POSIXct na coluna "Tempo" é igual a "2013-08-04" através da função subset() que é utilizada para selecionar as linhas com base numa condição lógica.

```
data_selecionada <-
subset(DADOS1,as.Date(DADOS1$Tempo) ==
as.Date("2013-08-04"))
```

Depois criamos outra variável chamada "temp_motores" a partir da variável "data_selecionada" que apenas irá conter as colunas "Tempo", "ESP01.3", "ESP02.3" e "ESP03.3".

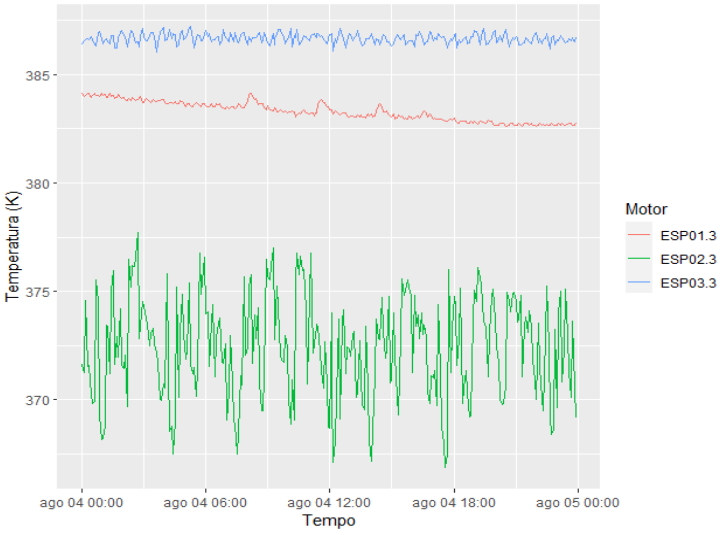
```
temp_motores <- data_selecionada[, c("Tempo",
"ESP01.3", "ESP02.3", "ESP03.3")]
```

Nesta linha, a função gather() é usada para transformar a variável "temp_motores". A função converte as colunas "ESP01.3", "ESP02.3" e "ESP03.3" em duas colunas: "Bomba" e "Temperatura". A coluna "Bomba" conterá os nomes das antigas colunas ("ESP01.3", "ESP02.3", "ESP03.3"), enquanto a coluna "Temperatura" conterá os valores correspondentes. O argumento -Tempo indica que a coluna "Tempo" deve ser mantida sem ser alterada.

```
temp_motores_final <- gather(temp_motores, key =
"Bomba", value = "Temperatura", -Tempo)
```

Em seguida geramos o gráfico com as funções ggplot, "aes()" define as variáveis onde "Tempo" é mapeado para o eixo x, "Temperatura" é mapeado para o eixo y, e "Bomba" é mapeado pela cor. Em seguida, geom_line() é chamado para adicionar as linhas no gráfico. labs() é usado para definir os rótulos dos eixos x e y, e o título da legenda da cor.

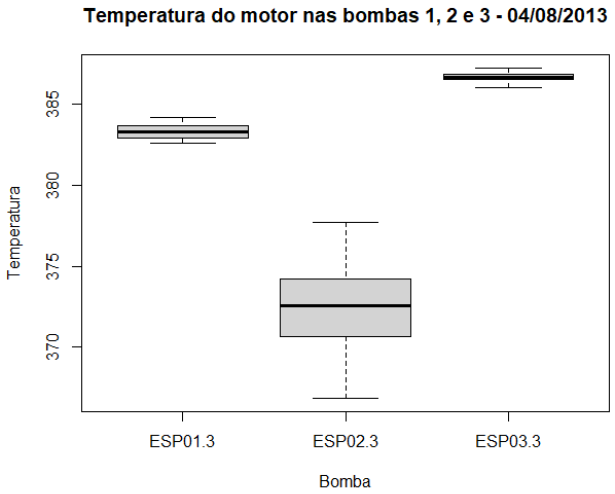
```
ggplot(data = temp_motores_final, aes(x = Tempo, y =
Temperatura, color = Bomba)) +
  geom_line() +
labs(x = "Tempo", y = "Temperatura (K)", color = "Motor")
```



C. *Alínea c)*

Neste ponto já temos todos os dados necessários só nos resta gerar o boxplot com a função *boxplot e axis*:

```
boxplot(temp_motores$ESP01.3,
temp_motores$ESP02.3, temp_motores$ESP03.3,
main="Temperatura do motor nas bombas 1, 2 e 3 -
04/08/2013", ylab="Temperatura", xlab="Bomba")

axis(1, at = c(1:3), labels = c("ESP01.3",
"ESP02.3","ESP03.3"))
```



Tendo em vista os resultados, pode-se observar que o motor com a temperatura mais alta é da bomba 3, e o motor com a temperatura mais baixa o da bomba 2. Também se pode verificar que o motor da bomba 2 tem a temperatura mais flutuante e o motor da bomba 1 tem a temperatura mais estável.

D. Alínea d)

i. Antes de mais vamos criar um novo dataframe apenas com os dados do mês de Março de 2014, com o auxílio da função *filter*:

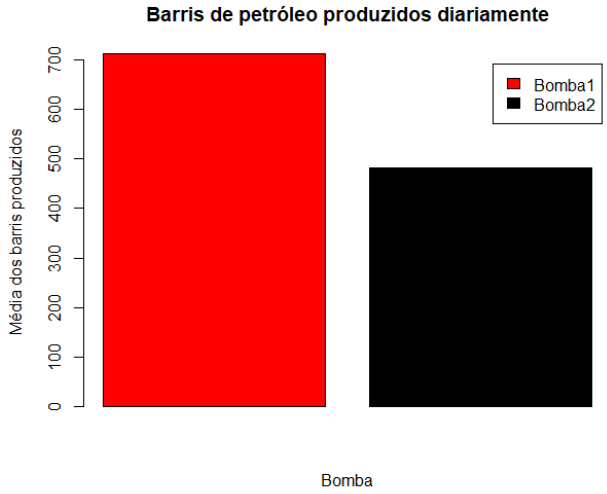
```
df_filtered2 <- filter(dados, as.Date(datetime)
>= as.Date("2014-03-01") & as.Date(datetime)
<= as.Date("2014-03-31"))
```

Depois criamos uma lista para cada uma das bombas com o *oil rate* de cada bomba e depois a média de cada dia, usando a função *substr* com o intervalo [1,7] para captarmos a parte da data em que refere o ano e o mês:

```
df_filtered2oil1 <-
as.numeric(dados[substr(dados$datetime, 1, 7)
== "2014-03", ]$IC01.8)

df_filtered2oil2 <-
as.numeric(DADOS1[substr(DADOS1$Tempo,
1, 7) == "2014-03", ]$IC02.8)


media_bomba1 <-
mean(na.omit(df_filtered2oil1))

media_bomba2 <-
mean(na.omit(df_filtered2oil2))
```

De seguida já podemos gerar o gráfico de barras que mostra a média diária de barris produzidos, usando a função *barplot* com as médias das duas bombas:

```
barplot(c(media_bomba1, media_bomba2),

main = "Barris de petróleo produzidos
diariamente",

ylab = "Média dos barris produzidos",

xlab = "Bomba",

col = cores,

legend.text = c("Bomba1", "Bomba2"))
```



Concluímos que a média de barris produzidos diariamente é superior na bomba1.

ii. Mais uma vez precisamos de criar um dataframe diferente desta vez com as datas definidas de 1 junho de 2013 até 31 maio de 2014:

```
intervalo_dados <- subset(DADOS1,
as.Date(Tempo) >= as.Date("2013-06-01") &
as.Date(Tempo) <= as.Date("2014-05-31") &
IC01.8)
```

De seguida calculamos a quantidade de petróleo de cada bomba por mês

```
quantidade_petroleo<- intervalo_dados
%>%

mutate(Mes = format(as.Date(Tempo),
"%Y-%m")) %>%

group_by(Mes) %>%

summarise(Barris = sum(`IC01.8`))
```

Por último, encontramos o mês em que a Bomba 1 extraiu mais barris de petróleo

```
mes_max <-
quantidade_petroleo$Mes[which.max(quantidad
e_petroleo$Barris)]
```

mes_max	"2014-03"

É criada a variável mes_max e observamos que o mês que a bomba 1 extraiu mais barris de petróleo foi o mês de Março de 2014.

iii. Para calcular a produção diária, nos dias da amostra aleatória, começamos por dar *set.seed(300)* como foi pedido no enunciado e inicializar duas listas para armazenarem a produção diária de cada bomba:

```
set.seed(300)

bomba1_daily_prod <- c()

bomba2_daily_prod <- c()
```

Depois fazemos um loop *for* que vai rodar todos os 10 elementos aleatórios da função *sample*. Dentro do loop vai ter mais duas inicializações de duas listas auxiliares para guardar os valores de cada dia. De seguida enquanto vamos rodando o loop vamos comparando com a lista o número do dia (usamos a fórmula (tempo-1370044800) /86400+1 para calcular o número do dia) e se der *match* usamos a função *append* para escrever na lista auxiliar. Quando tivermos todos os valores desse dia no vetor fazemos a média usando a função *mean* e damos *append* na lista inicializada antes do loop *for*:

```
for(i in sample(1:365,10)){

aux1 <- c()

aux2 <- c()


aux1<-append(aux1,
as.numeric(DADOS1[as.integer((as.numeric(D
ADOS1$Tempo_seg)-1370044800)/86400+1)
== i,]$IC01.8))

bomba1_daily_prod<-
append(bomba1_daily_prod,
mean(na.omit(aux1)))


aux2<-append(aux2,
as.numeric(DADOS1[as.integer((as.numeric(D
ADOS1$Tempo_seg)-1370044800)/86400+1)
== i,]$IC02.8))


bomba2_daily_prod<-
append(bomba2_daily_prod,
mean(na.omit(aux2)))
```

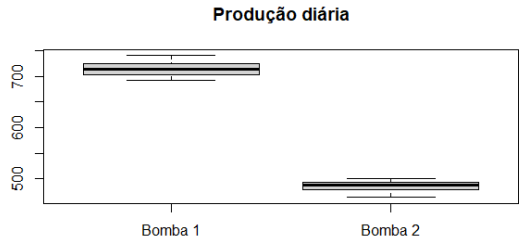
Posteriormente criamos outra dataframe para alojar as duas listas com a produtividade diária de cada bomba:

```
df_daily_prod<- list(bomba1_daily_prod,
bomba2_daily_prod)
```

Por último podemos gerar o boxplot, desta vez usando a função *xaxt* para suprimir a impressão de valores no eixo dos xx e usamos a função *axis* para escrever no nome de cada bomba no eixo dos xx:

```
boxplot(df_daily_prod, xaxt = "n", main =
"Produção diária")

axis(1, at = c(1, 2), labels = c("Bomba 1",
"Bomba 2"))
```

Analisando o boxplot podemos ver que a bomba 1 tem maior produção diária.



iv. Teste de Hipóteses

H0: μ1 <= μ2 - a média da produção diária de petróleo da bomba 1 é menor ou igual à média da produção diária de petróleo da bomba 2;

H1: μ1 > μ2 - a média da produção diária de petróleo da bomba 1 é maior do que a média da produção diária de petróleo da bomba 2.

Nível de significância α = 0,05.

Antes de tudo precisamos de fazer um teste de Shapiro-Wilk para verificar se as amostras seguem uma distribuição normal:

```
shapiro.test(bomba1_daily_prod)

shapiro.test(bomba2_daily_prod)
```

Como a distribuição da bomba 1 deu p-value = 0.2503 e a da bomba 2 deu p-value = 0.9907 (são valores acima do nível de significância 0.05) as duas distribuições são normais, logo podemos usar testes paramétricos e decidimos usar o *teste t:*

```
t.test(na.omit(bomba1_daily_prod),
na.omit(bomba2_daily_prod),
alternative="greater", conf.level = 0.95)
```

Como o p-value dá 0.00003584, é menor do que o nível de significância de α = 0,05, logo rejeitamos a hipótese nula H0 e concluímos que há evidencia estatística para suportar a hipótese alternativa H1. Ou seja, a média da produção diária de petróleo da bomba 1 é maior do que a

média da bomba 2 no período de 1-6-2013 e 31-05-2014.

v. Para analisarmos se os dados da amostra correspondem com a realidade vamos filtrar todos os dados no período determinado para avaliarmos se a média de produção diária na bomba 1 é maior que na bomba 2:

```
bomba1_oil_rate <- as.numeric(DADOS1[as.Date(DADOS1$Tempo) >= as.Date("2013-06-01") & as.Date(DADOS1$Tempo) <= as.Date("2014-05-31"), ]$IC01.8)

bomba2_oil_rate <- as.numeric(DADOS1[as.Date(DADOS1$Tempo) >= as.Date("2013-06-01") & as.Date(DADOS1$Tempo) <= as.Date("2014-05-31"), ]$IC02.8)
```

Finalmente damos print aos resultados e descobrimos que a média da bomba 1 é 711.5532 e a média da bomba 2 é 482.7037 ou seja podemos verificar que na "realidade" a hipótese alternativa está correta visto que a média de produção diária da bomba 1 é maior do que a média da bomba 2 no mesmo intervalo de tempo.

III. Exercicio 2

A. *Alínea a)*

Primeiramente importamos os dados com a função *read.csv*.

```
data <- read.csv("DADOS2.csv", sep =",")
```

De seguida verificamos a normalidade da distribuição dos dados, onde H0: distribuição normal e H1: distribuição não normal. Como, n<30 são feitos testes de Shapiro

```
shapiro.test(data$SVM) # p-value = 0.2687
shapiro.test(data$DT)  # p-value = 0.06772
shapiro.test(data$KN)  # p-value = 0.6926
shapiro.test(data$RF)  # p-value = 0.3138
shapiro.test(data$ML)  # p-value = 0.02138
shapiro.test(data$GB)  # p-value = 0.5125
```

Como existe pelos menos um conjunto de dados com p-value < 0.05, o teste é não conclusivo, logo serão feitos testes com os coeficientes de Pearson e de Spearman

```
cor_matrix_pearson <- rcorr(as.matrix(data[3:8]), type = "pearson")
```

	SVM	DT	KN	RF	ML	GB
SVM	1.0000000	0.2619970	0.6374486	0.4659169	0.7111270	0.8629016
DT	0.2619970	1.0000000	0.4339395	0.8802559	0.6247459	0.2127059
KN	0.6374486	0.4339395	1.0000000	0.4834416	0.8539377	0.7502013
RF	0.4659169	0.8802559	0.4834416	1.0000000	0.5719541	0.3240401
ML	0.7111270	0.6247459	0.8539377	0.5719541	1.0000000	0.7211135
GB	0.8629016	0.2127059	0.7502013	0.3240401	0.7211135	1.0000000

A correlação entre SVM e os outros métodos é geralmente positiva, com valores variando de moderados a fortes (0,46 a 0,86). Isso sugere que há uma relação razoável entre os resultados do SVM e os outros métodos considerados.

A correlação entre DT e RF é alta (0,88), indicando uma forte relação entre esses dois métodos. Isso ocorre porque o Random Forest(RF) é uma extensão do Decision Tree(DT), portanto, é esperado que sejam altamente correlacionados.

A correlação entre DT e ML é moderada (0,62), sugerindo uma relação razoável entre esses dois métodos.

A correlação entre GB e SVM é bastante alta (0,86), indicando uma forte relação entre esses dois métodos.

A correlação entre KN e ML é alta (0,85), sugerindo uma forte relação entre esses dois métodos.

n=10

	SVM	DT	KN	RF	ML	GB
SVM	NA	0.4646280709	0.047413448	0.1747246831	0.021123270	0.001305594
DT	0.464628071	NA	0.210213438	0.0007765737	0.053467750	0.555192125
KN	0.047413448	0.2102134376	NA	0.1569020144	0.001663052	0.012441036
RF	0.174724683	0.0007765737	0.156902014	NA	0.084062910	0.361019768
ML	0.021123270	0.0534677502	0.001663052	0.0840629096	NA	0.018596995
GB	0.001305594	0.5551921245	0.012441036	0.3610197680	0.018596995	NA

A matriz de valores p, também conhecida como matriz de significância os valores p representam a probabilidade de obter uma correlação igual ou mais extrema entre os métodos, assumindo que não há correlação real na população. Essa matriz de valores p é utilizada para testar a hipótese nula de que não há correlação entre os métodos. Um valor p baixo (geralmente abaixo de um limiar de significância, como 0,05) indica que a correlação é estatisticamente significativa, ou seja, é improvável que tenha ocorrido apenas por acaso. Por outro lado, um valor p alto sugere que a correlação observada pode ser explicada por variações aleatórias e não é estatisticamente significativa. Posto isto, é possível retirar algumas conclusões:

A correlação entre SVM e DT apresenta um valor p de 0.4646, indicando que essa correlação não é estatisticamente significativa a um nível de significância de 0.05. Isso significa que a relação entre esses dois métodos pode ser explicada por variações aleatórias.

A correlação entre SVM e KN também não é estatisticamente significativa, com um valor p de 0.0474.

A correlação entre SVM e RF possui um valor p de 0.1747, indicando uma correlação não significativa a um nível de significância de 0.05.

A correlação entre SVM e ML apresenta um valor p de 0.0211, indicando uma correlação estatisticamente significativa.

A correlação entre SVM e GB apresenta um valor p muito baixo de 0.0013, indicando uma correlação estatisticamente significativa e robusta entre esses dois métodos.

A correlação entre DT e KN possui um valor p de 0.2102, sugerindo uma correlação não significativa.

A correlação entre DT e RF é estatisticamente significativa, com um valor p muito baixo de 0.0008.

A correlação entre DT e ML possui um valor p de 0.0535, indicando uma correlação estatisticamente significativa.

A correlação entre DT e GB é estatisticamente significativa, com um valor p de 0.5552.

A correlação entre KN e RF também é estatisticamente significativa, com um valor p de 0.1569.

A correlação entre KN e ML é estatisticamente significativa, com um valor p de 0.0017.

A correlação entre KN e GB possui um valor p de 0.0124, indicando uma correlação estatisticamente significativa.

A correlação entre RF e ML possui um valor p de 0.0841, indicando uma correlação estatisticamente significativa.

A correlação entre RF e GB é estatisticamente significativa, com um valor p de 0.3610.

A correlação entre ML e GB é estatisticamente significativa, com um valor p de 0.0186.

Fizemos o mesmo processo para spearman.

cor_matrix_spearman <- rcorr(as.matrix(data[3:8]), type = "spearman")

	SVM	DT	KN	RF	ML	GB
SVM	1.0000000	0.3671548	0.7284089	0.6584896	0.6851982	0.8473561
DT	0.3671548	1.0000000	0.5033039	0.8435960	0.6228386	0.2127252
KN	0.7284089	0.5033039	1.0000000	0.5474032	0.9018405	0.7616428
RF	0.6584896	0.8435960	0.5474032	1.0000000	0.7033672	0.4475650
ML	0.6851982	0.6228386	0.9018405	0.7033672	1.0000000	0.6749517
GB	0.8473561	0.2127252	0.7616428	0.4475650	0.6749517	1.0000000

A correlação entre SVM e os outros métodos é geralmente positiva, com valores variando de moderados a fortes. Por exemplo, a correlação entre SVM e DT é 0.367, SVM e KN é 0.728, SVM e RF é 0.658, SVM e ML é 0.685 e SVM e GB é 0.847.

A correlação entre DT e RF é alta, com um valor de 0.843, indicando uma forte relação entre esses dois métodos.

A correlação entre KN e ML é alta, com um valor de 0.902, o que sugere uma relação forte entre esses dois métodos.

A correlação entre GB e SVM também é alta, com um valor de 0.847, indicando uma forte relação entre eles.

As correlações entre DT e KN, DT e ML, RF e ML, RF e GB, e ML e GB são moderadas, com valores de correlação entre 0.5 e 0.7.

n=10

	SVM	DT	KN	RF	ML	GB
SVM	NA	0.296640036	0.0168895167	0.038423402	0.0287684137	0.001967175
DT	0.296640036	NA	0.1380843619	0.002157946	0.0544188915	0.555155433
KN	0.016889517	0.138084362	NA	0.101461335	0.0003602558	0.010470093
RF	0.038423402	0.002157946	0.1014613350	NA	0.0232427539	0.194622217
ML	0.028768414	0.054418892	0.0003602558	0.023242754	NA	0.032249438
GB	0.001967175	0.555155433	0.0104700930	0.194622217	0.0322494384	NA

>

A maioria das correlações entre os métodos apresenta valores p baixos, indicando que as correlações são estatisticamente significativas. Isso sugere que as correlações observadas não ocorreram apenas por acaso, mas são indicativas de uma relação real entre os métodos.

A correlação entre SVM e DT possui um valor p de 0.2966, indicando que essa correlação não é estatisticamente significativa a um nível de significância de 0.05. Isso sugere que a relação entre esses dois métodos pode ser explicada por variações aleatórias.

A correlação entre SVM e KN possui um valor p de 0.0169, o que indica que essa correlação é estatisticamente significativa.

A correlação entre SVM e RF também é estatisticamente significativa, com um valor p de 0.0384.

A correlação entre SVM e ML é estatisticamente significativa, com um valor p de 0.0288.

A correlação entre SVM e GB possui um valor p muito baixo de 0.0019, indicando uma correlação estatisticamente significativa e robusta entre esses dois métodos.

A correlação entre DT e KN é estatisticamente significativa, com um valor p de 0.1381.

A correlação entre DT e RF também é estatisticamente significativa, com um valor p muito baixo de 0.0022.

A correlação entre DT e ML é estatisticamente significativa, com um valor p de 0.0544.

A correlação entre DT e GB possui um valor p de 0.5552, o que indica que essa correlação não é estatisticamente significativa.

A correlação entre KN e RF também é estatisticamente significativa, com um valor p de 0.1015.

A correlação entre KN e ML é estatisticamente significativa, com um valor p de 0.0004.

A correlação entre KN e GB também é estatisticamente significativa, com um valor p de 0.0105.

A correlação entre RF e ML é estatisticamente significativa, com um valor p de 0.0232.

A correlação entre RF e GB também é estatisticamente significativa, com um valor p de 0.1946.

A correlação entre ML e GB é estatisticamente significativa, com um valor p de 0.0322.

B. Alínea b)

Nesta alínea avaliamos a Homogeneidade de variâncias e começamos por criar uma variável data_join para dar Melt aos dados

data_join <- melt(data[3:8], variable.name = "Type", value.name = "Precision")

Seguimos for formular hipóteses, onde

H0: existe homogeneidade entre variâncias

H1: não existe homogeneidade entre variâncias dos grupos.

Realizamos o teste de Levene

leveneTest(Precision ~ Type, data_join, center = mean)

```
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value    Pr(>F)
group  5  6.2254 0.0001259 ***
      54
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Quando o valor p é menor que 0,05 (ou seja, abaixo de um nível de significância de 5%), rejeitamos a hipótese nula, logo não podemos admitir a existência de homogeneidade entre variâncias.

Portanto, como o teste ANOVA falha, teremos que realizar teste não paramétrico.

C. Alínea c)

Aqui efetuamos teste de Friedman já que as amostras são emparelhadas. Formulamos novamente novas hipóteses, em que H0 Formulação de hipóteses: não existem diferenças significativas entre a precisão dos algoritmos e H1: existem diferenças significativas entre a precisão dos algoritmos.

```
data_friedman <- data.frame(dataSet =
rep(DADOS2$dsets, 6), algoritmo = c(rep(
"SVM", 10),rep("DT", 10),rep("KN",10),rep("RF",
10),rep("ML",10),rep("GB", 10)), result =
c(DADOS2$SVM, DADOS2$DT, DADOS2$KN,
DADOS2$RF, DADOS2$ML, DADOS2$GB))
```

```
friedman.test(result~algoritmo|dataSet, dados =
data_friedman)
```

Como p-value>0.05, não rejeitamos H0, logo é possível afirmar que que não existem diferenças significativas entre a precisão dos diferentes algoritmos.

IV. Exercicio 3

A. Alínea a)

Importamos os dados com a função read.csv e atribuímos à variável DADOS3.

```
DADOS3 <- read_csv("Dados excel/DADOS3.csv")
```

Depois criamos três variáveis separadas para cada grupo de cilindros: Para isto vamos criar 3 subsets e atribuí-los às variáveis dados_4cil, dados_6cil e dados_8cil, respetivamente.

```
dados_4cil <- subset(DADOS3, Cylinders == 4)
dados_6cil <- subset(DADOS3, Cylinders == 6)
dados_8cil <- subset(DADOS3, Cylinders == 8)
```

Depois criamos um dataset onde juntamos os 3 subsets.

```
dataset <- c (dados_8cil, dados_6cil, dados_4cil)
```

Para garantir que o teste de Kruskal-Wallis é realizado corretamente, é necessário criar uma variável categórica chamada "groups" que indica a qual grupo cada cilindro pertence. Essa variável "groups" servirá como um limite para diferenciar os diferentes grupos durante o teste.

```
groups <-
factor(c(rep("Eight",length(dados_8cil)),rep("Six",length(da
dos_6cil)),rep("Four",length(dados_4cil))))
```

Depois estabelecemos as duas hipóteses H0 (hipótese nula) e H1 (hipótese alternativa).

```
h0 <- "H0: "Não há diferenças significativas entre a
aceleração dos três grupos."
h1 <- "H1: Há diferenças significativas entre a aceleração
dos três grupos."
```

De seguida, iremos fazer o teste de Shapiro para determinar se a amostra de dados provém de uma distribuição normal.

```
shapiro.test(DADOS3$Cylinders)
```

```
        Shapiro-Wilk normality test

data:  DADOS3$Cylinders
W = 0.71974, p-value = 1.87e-12

>
```

O p-value do teste de Shapiro deu inferior a 5% logo a amostra de dados não provém de uma distribuição normal.

Por isso, devemos usar testes não paramétricos, logo o único teste estudado que faz esta verificação entre grupos de mais de dois elementos é o Kruskal – Wallis.

Teste Kruskal – Wallis e atribuímos a uma variável “results”.

```
results <- kruskal.test(dataset,groups)
```

```
        Kruskal-Wallis rank sum test

data:  dataset
Kruskal-Wallis chi-squared = 387.53, df = 11, p-value < 2.2e-16
```

Depois temos uma if condition em que se o p-value do teste de Kruskal – Wallis for inferior a 5% então podemos afirmar que é lógico rejeitar H0. Se for superior a 5% então H0 não pode ser descartado.

```
alpha <- 0.05

if(results$p.value<=alpha){

print("The p-value is lower to 5%, therefore is logical to
reject H0")

}else{

print("The p-value is higher to 5%, therefore H0 cannot
be discarted")

}
```

```
[1] "H0 rejeitada. Pelo menos um grupo é estatisticamente diferente."
>
```

Conseguimos então, com o resultado do teste, concluir que é possível rejeitar H0, uma vez que a confiança disponibilizada pelo p-value não é elevada o suficiente.

Concluímos assim que existem diferenças significativas entre a aceleração de diversos veículos, tendo em consideração apenas o seu número de cilindros.

B. Alínea b)

i)

Começamos por encontrar o modelo de regressão linear, tendo em conta que a aceleração era a variável dependente.

```
linearReg <- lm(Acceleration ~ Horsepower + Weight +
Cylinders, data = DADOS3)
```

```
 (Intercept)   Horsepower      Weight   Cylinders
18.424687436 -0.065715379  0.003339054 -1.088690330
> linearReg[["call"]]
lm(formula = Acceleration ~ Horsepower + Weight + Cylinders,
    data = DADOS3)
>
```

ii)

Depois, usamos este modelo de regressão linear para estimar a aceleração de uma viatura com um peso de 2950 kg, potência de 100 Hp e 4 cilindros.

Para isto criamos um dataframe com estes valores.

```
new <- data.frame(Horsepower = 100, Weight= 2950,
Cylinders = 4)
```

De seguida, utilizamos a função ‘predict’ para estimar a aceleração da viatura.

```
prediction <- predict(linearReg,newdata = new)
```

```
names(prediction) <- "Acceleration"
```

Com isto, chegamos a um valor aproximado de 17.3 para a aceleração da viatura.

Values	
prediction	Named num 17.3

V. Conclusão

A. Conclusão

Este trabalho fornece uma análise estatística detalhada do conjunto de dados usando a ferramenta R. Os resultados obtidos demonstram a eficiência do R na realização de análises estatísticas complexas.

Este trabalho também nos ensinou como usar efetivamente a linguagem de programação R para explorar, manipular e analisar grandes conjuntos de dados.

A capacidade de adaptar a análise às necessidades específicas também é uma vantagem importante do uso do R. No entanto, é importante salientar que é necessário conhecimento estatístico suficiente para a correta interpretação dos resultados, o que reforça a importância de um bom conhecimento teórico.

Graças à perseverança e ao espírito cooperativo, conseguimos superar esses obstáculos e dar continuidade a esse projeto. Partilhar conhecimento e a ajuda mútua foram a chave para alcançar os nossos objetivos e concluir as tarefas com sucesso. Acreditamos que este processo colaborativo não só melhora o produto final, mas também contribui para o desenvolvimento pessoal e profissional de cada membro.