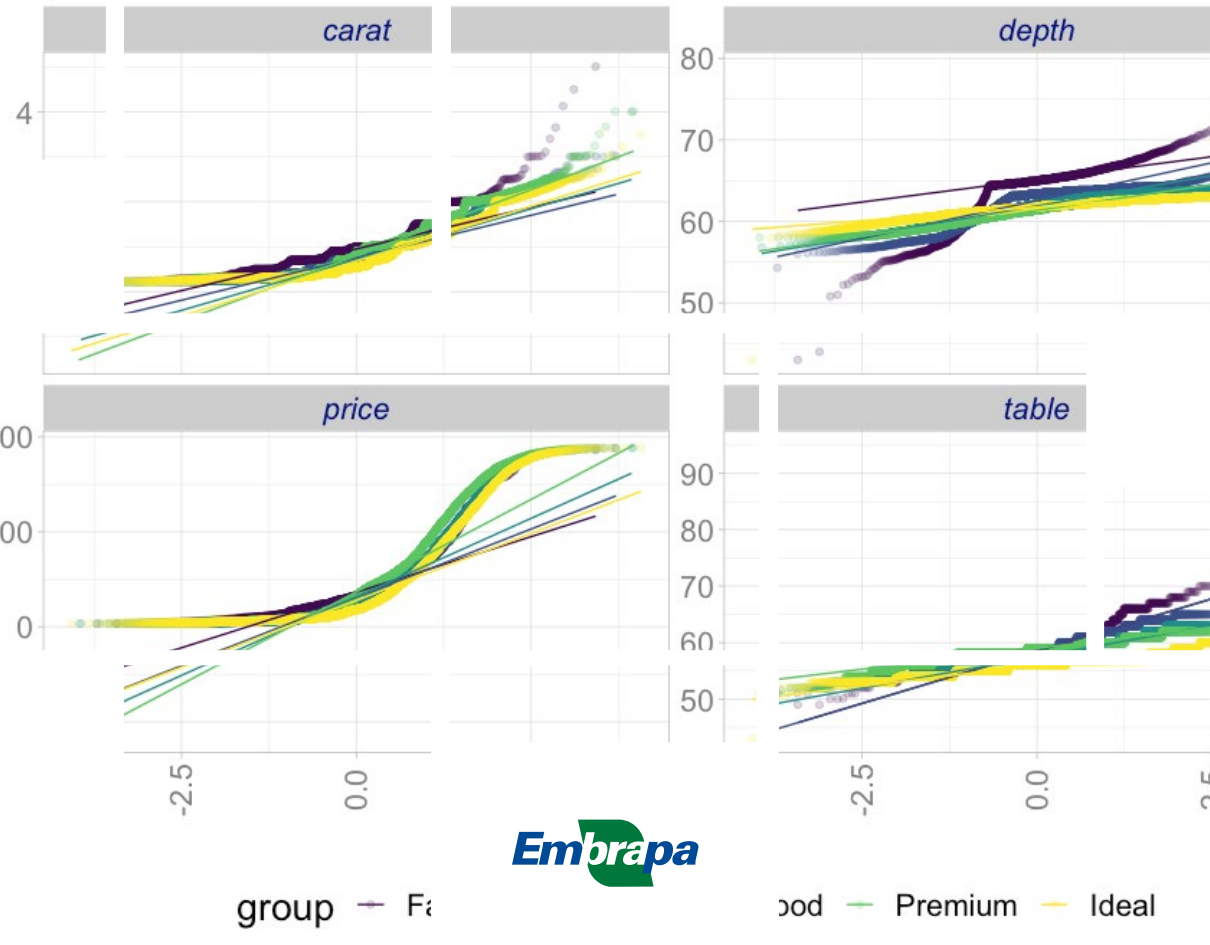
DOCUMENTOS
74

ISSN 1981-6103
Novembro, 2023

Introdução à análise exploratória de dados com R



ISSN 1981-6103
Novembro, 2023

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Roraima
Ministério da Agricultura e Pecuária

DOCUMENTOS 74

Introdução à análise exploratória de dados com R

George Amaro

Embrapa Roraima
Boa Vista, RR
2023

Exemplares desta publicação podem ser obtidos na:

Embrapa Roraima
Rod. BR-174 Km 08 - Distrito Industrial Boa Vista-RR
Caixa Postal 133.
69301-970 - Boa Vista - RR
Telefax: (095) 3626-7018
e-mail: sac@cpafrr.embrapa.br
www.cpafrr.embrapa.br

Revisão editorial
Jeana Garcia Beltrão Macieira

Normalização Bibliográfica
Jeana Garcia Beltrão Macieira

Revisão de texto
Ilda Maria Sobral de Almeida

Editoração eletrônica
Phábrica de Produções:
Alecsander Coelho, Daniela Bissiguini,
Érsio Ribeiro e Paulo Ciola

Foto de capa
George Amaro

1ª edição
1ª impressão (2022): 200 exemplares

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Roraima

Amaro, George.
Introdução à análise exploratória de dados com R / George Amaro. – Boa Vista, RR : Embrapa Roraima, 2023.
35 p. : il. color. - (Documentos / Embrapa Roraima, ISSN 1981-6103; 74).

1. Técnicas estatística. 2. Data science. 3. Análise de dados. I. I. Título. II. Série.

CDD 519

Jeana Garcia Beltrão Macieira (CRB-11/589)

Autor

George Amaro
Administrador, mestre em Economia, pesquisador da Embrapa Roraima, Boa Vista, RR

Agradecimentos

À Empresa Brasileira de Pesquisa Agropecuária (Embrapa), cujo suporte financeiro, através do projeto “Avaliação da Distribuição Geográfica e Riscos Econômicos Potenciais da Mosca-Oriental-das- Frutas (*Bactrocera dorsalis*) e da Mosca-da-Carambola (*Bactrocera carambolae*) no Brasil.” (10.20.03.056.00.00), desenvolvido na Embrapa Roraima, possibilitou a realização deste trabalho.

Aos revisores anônimos que ofereceram valiosas contribuições para melhoria do texto. E a toda equipe de suporte da Embrapa Roraima, pelo apoio recebido durante a realização das atividades necessárias, especialmente à bibliotecária da Embrapa Roraima, Jeana Garcia Beltrão Macieira, pela normalização bibliográfica.

Sumário

Introdução à análise exploratória de dados com R

Introdução

A Análise Exploratória de Dados (AED) tem como finalidade realizar uma investigação inicial sobre os dados, resumindo suas características através de técnicas estatísticas e de visualização, e é uma etapa inicial crítica em qualquer fluxo de trabalho de *Data Science*, para descobrir padrões, detectar anomalias, testar hipóteses fundamentais e verificar suposições. Uma boa análise exploratória permite entender os dados, inicialmente, além de possibilitar *insights* e sedimentar a escolha das técnicas e ferramentas para as análises mais detalhadas, posteriormente.

Há pelo menos três motivações para analisar dados (Pearson, 2018):

1) entender o que aconteceu ou o que está acontecendo;

2) prever o que provavelmente acontecerá, seja no futuro ou em outras circunstâncias desconhecidas;

3) fomentar a tomada de decisões.

Nesse sentido, o foco da AED é a primeira motivação: entender os dados. De forma geral, “nós olhamos para números ou gráficos e tentamos encontrar padrões. Buscamos pistas sugeridas por informações básicas, imaginação, padrões percebidos e experiência com outras análises de dados” (Diaconis, 2006).

Naturalmente, o objetivo deste trabalho não é abordar detalhes da utilização do ambiente R e nem tampouco oferecer uma abordagem detalhada das possibilidades da AED com R, mas sim apresentar algumas opções para realizar a análise exploratória de forma automatizada, com o uso de pacotes desenvolvidos para esse fim.

É importante ainda ressaltar que existem gráficos exploratórios e explanatórios (Iliinsky; Steele, 2011). Ou seja, técnicas que são úteis para realizar a análise dos dados e que permitem avaliar a sua estrutura, relações, ten-

dências e discrepâncias, não necessariamente são adequadas para explicar e comunicar essas descobertas. Dessa forma, também está fora do escopo deste trabalho a preocupação e o esmero estético das visualizações apresentadas, necessários à sua publicação ou apresentação, embora alguns recursos sejam utilizados, principalmente a título de ilustração das possibilidades.

Exemplos de Análises Exploratórias

Para ilustrar a utilização de alguns pacotes para análise exploratória automatizada de dados, foi utilizado o ambiente R versão 4.2.1[1] (R Core Team, 2022), o conjunto de dados *diamonds* (Tabela 1) do pacote *{ggplot2}* (Wickham, 2016), que é utilizado em vários exemplos de visualização e análise de dados e está disponível juntamente com o pacote.

Tabela 1. Descrição das variáveis do conjunto de dados *diamonds*, utilizado para exemplificar as análises e utilização dos pacotes.

Variável	Descrição	Valores
price	Preço em US$	326 – 18.823
carat	Peso do diamante	0,2 – 5,01
cut	Qualidade do corte	Fair, Good, Very Good, Premium, Ideal
color	Cor do diamante	J (pior) – D (melhor)
clarity	Medida de claridade	I1 (pior), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (melhor)
x	Comprimento em mm	0 – 10,74
y	Largura em mm	0 – 58,9
z	Profundidade em mm	0 – 31,8
depth	Percentual da profundidade total	43 – 79
table	Largura do topo do diamante em relação ao ponto mais largo	43 – 95

[1] Os trechos de código (*snippets*) apresentados foram formatados com a utilização do *Syntax Highlight Code In Word Documents*, disponível em https://planetb.troye.io.

Inicialmente, deve-se carregar o conjunto de dados:

```
data(diamonds, package = "ggplot2")
```

Pode-se, a partir de funções básicas do R, obter informações sobre os dados, como sua estrutura interna (*str*), uma amostra de conteúdo das primeiras linhas (*head*) e estatísticas descritivas básicas (*summary*):

```
str(diamonds)
```

```
tibble [53,940 × 10] (S3: tbl_df/tbl/data.frame)
 $ carat  : num [1:53940] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
 $ depth  : num [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
head(diamonds)
```

```
# A tibble: 6 × 10
  carat cut       color clarity depth table price     x     y     z
  <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
summary(diamonds)
```

```
     carat               cut        color        clarity          depth           table      
 Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00   Min.   :43.00  
 1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00   1st Qu.:56.00  
 Median :0.7000   Very Good :12082   F: 9542   SI2     : 9194   Median :61.80   Median :57.00  
 Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75   Mean   :57.46  
 3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50   3rd Qu.:59.00  
 Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00   Max.   :95.00  
                                    J: 2808   (Other): 2531                                  
     price             x                y                z          
 Min.   :  326   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 2401   Median : 5.700   Median : 5.710   Median : 3.530  
 Mean   : 3933   Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
 3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :18823   Max.   :10.740   Max.   :58.900   Max.   :31.800  
```

Essa exploração preliminar permite determinar se os dados que temos são os que esperávamos ter. Isso é especialmente importante considerando o volume atual de dados que normalmente são analisados, que os dados podem ter sido coletados ou organizados por pessoas com as quais o pesquisador tenha pouco ou mesmo nenhum contato direto ou mesmo que tenham sido adquiridos por equipamentos automáticos ou através da Internet ou uso de Apps em dispositivos móveis.

Os pacotes *{gtsummary}* e *{vtable}*

Visualizar os dados na forma de uma tabela analítica se constitui uma poderosa ferramenta e ainda é extremamente útil para publicação de resultados de pesquisas, possibilitando caracterizar as variáveis através de diversos indicadores.

Dentre os vários pacotes que implementam essa função no R, dois destacam-se por sua simplicidade de uso e capacidade de customização: *{gtsummary}*[2] (Sjoberg et al., 2021) e *{vtable}*[3] (Huntington-Klein, 2022).

De acordo com a sua descrição, o pacote *{gtsummary}* fornece uma maneira elegante e flexível de criar tabelas analíticas e de resumo prontas para publicação usando a linguagem de programação R, resumindo conjuntos de dados, modelos de regressão, entre outros, com recursos altamente personalizáveis.

Para gerar a Tabela 2, foi utilizado o seguinte trecho de código, onde as variáveis contínuas são apresentadas com suas médias e desvios-padrão e as variáveis categóricas com suas frequências absolutas e relativas, agrupadas pela qualidade do corte (*cut*). A definição de agrupamentos pode ser por qualquer variável, conforme a necessidade da análise.

[2] Versão 1.6.1.

[3] Versão 1.3.4.

```
# gtsummary
install.packages("gtsummary")        # se necessário
library(gtsummary)

table <- tbl_summary(
  data = diamonds,
  by = cut,
  missing = "no",
  statistic = list(all_continuous() ~ "{mean} ({sd})",
                   all_categorical() ~ "{n} ({p}%)"),
  digits = all_continuous() ~ 2,
) %>%
  add_n() %>%
  add_p() %>%
  modify_header(label = "**Variáveis**") %>%
  bold_labels()

table
```

Pode ser necessário instalar o pacote *{gtsummary}* (linha 2 do *snippet* acima). O operador ***%>%*** (*forward pipe*) cria um encadeamento de comandos: o resultado de uma expressão é passado para a próxima, como nas linhas 12 a 15, para adicionar elementos à tabela criada entre as linhas 5 e 12.

As tabelas criadas também podem ser salvas diretamente como arquivos de imagem, HTML, RTF, LaTeX ou Word, através da utilização de funções que propiciam uma interface com outros pacotes que possuem essas funcionalidades.

Tabela 2. Tabela gerada e formatada com o pacote *{gtsummary}* a partir do conjunto de dados *diamonds*.

Variáveis	N	**Fair**, N = 1,610[1]	**Good**, N = 4,906[1]	**Very Good**, N = 12,082[1]	**Premium**, N = 13,791[1]	**Ideal**, N = 21,551[1]	p-value[2]
carat	53,940	1.05 (0.52)	0.85 (0.45)	0.81 (0.46)	0.89 (0.52)	0.70 (0.43)	<0.001
color	53,940						<0.001
D		163 (10%)	662 (13%)	1,513 (13%)	1,603 (12%)	2,834 (13%)	
E		224 (14%)	933 (19%)	2,400 (20%)	2,337 (17%)	3,903 (18%)	
F		312 (19%)	909 (19%)	2,164 (18%)	2,331 (17%)	3,826 (18%)	
G		314 (20%)	871 (18%)	2,299 (19%)	2,924 (21%)	4,884 (23%)	
H		303 (19%)	702 (14%)	1,824 (15%)	2,360 (17%)	3,115 (14%)	
I		175 (11%)	522 (11%)	1,204 (10.0%)	1,428 (10%)	2,093 (9.7%)	
J		119 (7.4%)	307 (6.3%)	678 (5.6%)	808 (5.9%)	896 (4.2%)	
clarity	53,940						<0.001
I1		210 (13%)	96 (2.0%)	84 (0.7%)	205 (1.5%)	146 (0.7%)	
SI2		466 (29%)	1,081 (22%)	2,100 (17%)	2,949 (21%)	2,598 (12%)	
SI1		408 (25%)	1,560 (32%)	3,240 (27%)	3,575 (26%)	4,282 (20%)	
VS2		261 (16%)	978 (20%)	2,591 (21%)	3,357 (24%)	5,071 (24%)	
VS1		170 (11%)	648 (13%)	1,775 (15%)	1,989 (14%)	3,589 (17%)	
VVS2		69 (4.3%)	286 (5.8%)	1,235 (10%)	870 (6.3%)	2,606 (12%)	
VVS1		17 (1.1%)	186 (3.8%)	789 (6.5%)	616 (4.5%)	2,047 (9.5%)	
IF		9 (0.6%)	71 (1.4%)	268 (2.2%)	230 (1.7%)	1,212 (5.6%)	
depth	53,940	64.04 (3.64)	62.37 (2.17)	61.82 (1.38)	61.26 (1.16)	61.71 (0.72)	<0.001
table	53,940	59.05 (3.95)	58.69 (2.85)	57.96 (2.12)	58.75 (1.48)	55.95 (1.25)	<0.001
price	53,940	4,358.76 (3,560.39)	3,928.86 (3,681.59)	3,981.76 (3,935.86)	4,584.26 (4,349.20)	3,457.54 (3,808.40)	<0.001
x	53,940	6.25 (0.96)	5.84 (1.06)	5.74 (1.10)	5.97 (1.19)	5.51 (1.06)	<0.001
y	53,940	6.18 (0.96)	5.85 (1.05)	5.77 (1.10)	5.94 (1.26)	5.52 (1.07)	<0.001
z	53,940	3.98 (0.65)	3.64 (0.65)	3.56 (0.73)	3.65 (0.73)	3.40 (0.66)	<0.001

[1] Mean (SD); n (%)
[2] Kruskal-Wallis rank sum test; Pearson's Chi-squared test

O pacote *{vtable}* não tem a mesma flexibilidade do *{gtsummary}* mas consegue, de uma forma rápida e elegante, gerar um sumário de um conjunto de dados e apresentar o resultado em uma tabela formatada que é de fácil utilização. Embora não seja muito extensivo, existe um conjunto de opções de customização. Contudo, conforme o próprio autor do pacote, esse não é seu propósito, mas sim gerar uma tabela com o mínimo esforço necessário.

Com a execução do *snippet* abaixo, foi gerada a Tabela 3, com o uso da função *st* (uma abreviação de *sumtable*), após ser instalado e carregado o pacote.

```
# vtable
install.packages("vtable")
library(vtable)

st(diamonds)
```

Tabela 3. Tabela gerada e formatada com o pacote *{vtable}* a partir do conjunto de dados *diamonds*.

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
carat	53940	0.798	0.474	0.2	0.4	1.04	5.01
cut	53940						
... Fair	1610	3%					
... Good	4906	9.1%					
... Very Good	12082	22.4%					
... Premium	13791	25.6%					
... Ideal	21551	40%					
color	53940						
... D	6775	12.6%					
... E	9797	18.2%					
... F	9542	17.7%					
... G	11292	20.9%					
... H	8304	15.4%					
... I	5422	10.1%					
... J	2808	5.2%					
clarity	53940						
... I1	741	1.4%					
... SI2	9194	17%					
... SI1	13065	24.2%					

... VS2	12258	22.7%					
... VS1	8171	15.1%					
... VVS2	5066	9.4%					
... VVS1	3655	6.8%					
... IF	1790	3.3%					
depth	53940	61.749	1.433	43	61	62.5	79
table	53940	57.457	2.234	43	56	59	95
price	53940	3932.8	3989.44	326	950	5324.25	18823
x	53940	5.731	1.122	0	4.71	6.54	10.74
y	53940	5.735	1.142	0	4.72	6.54	58.9
z	53940	3.539	0.706	0	2.91	4.04	31.8

Dessa forma, quer pela utilização da função básica *summary*, ou pelo uso de pacotes como *{gtsummary}* e *{vtable}*, pode-se avaliar as características gerais de um conjunto de dados bem como algumas de suas estatísticas descritivas, o que, de forma geral, se constitui na primeira parte da AED.

O pacote *{tableone}*

Embora esse tenha pacote tenha sido desenvolvido pensando nas publicações da área biomédica e na apresentação dos resultados no padrão “Table 1”, como são normalmente vistos, sua utilização estende-se para qualquer área. O *{tableone}* (Yoshida; Bartel, 2022) contém muitas funções úteis para resumir dados categóricos e contínuos. As informações resultantes não são formatadas, mas podem ser mais bem organizadas com a função *print* e suas opções. Segue um exemplo dos resultados obtidos com execução do trecho de código abaixo:

```
# tableone
install.packages("tableone")
library(tableone)

table <- CreateTableOne(
  data = diamonds,
)

summary(table)
```

Cujo resultado é:

```
    ### Summary of continuous variables ###

strata: Overall
          n miss p.miss    mean      sd median    p25  p75    min    max  skew kurt
carat 53940    0      0     0.8     0.5    0.7    0.4    1    0.2      5  1.12  1.3
depth 53940    0      0    61.7     1.4   61.8   61.0   62   43.0     79 -0.08  5.7
table 53940    0      0    57.5     2.2   57.0   56.0   59   43.0     95  0.80  2.8
price 53940    0      0 3932.8  3989.4 2401.0  950.0 5324  326.0  18823  1.62  2.2
x     53940    0      0    5.7     1.1    5.7    4.7    7    0.0     11  0.38 -0.6
y     53940    0      0    5.7     1.1    5.7    4.7    7    0.0     59  2.43 91.2
z     53940    0      0    3.5     0.7    3.5    2.9    4    0.0     32  1.52 47.1

================================================================================

     ### Summary of categorical variables ###

strata: Overall
     var     n miss p.miss      level  freq percent cum.percent
     cut 53940    0    0.0       Fair  1610     3.0         3.0
                             Good  4906     9.1        12.1
                        Very Good 12082    22.4        34.5
                          Premium 13791    25.6        60.0
                            Ideal 21551    40.0       100.0

   color 53940    0    0.0          D  6775    12.6        12.6
                                    E  9797    18.2        30.7
                                    F  9542    17.7        48.4
                                    G 11292    20.9        69.3
                                    H  8304    15.4        84.7
                                    I  5422    10.1        94.8
                                    J  2808     5.2       100.0

 clarity 53940    0    0.0         I1   741     1.4         1.4
                                   SI2  9194    17.0        18.4
                                   SI1 13065    24.2        42.6
                                   VS2 12258    22.7        65.4
                                   VS1  8171    15.1        80.5
                                  VVS2  5066     9.4        89.9
                                  VVS1  3655     6.8        96.7
                                    IF  1790     3.3       100.0
```

O pacote *{DataExplorer}*

O *{DataExplorer}*[4] simplifica e automatiza o processo de AED e a geração de relatórios, fornecendo o perfil de cada variável do conjunto de dados, além oferecer várias funções úteis para gerar gráficos diferentes a partir de variáveis discretas e contínuas. O pacote permite gerar um relatório completo, em formato HTML de forma rápida e prática, apenas invocando a função *create_report* sobre um conjunto de dados. É possível passar argumentos adicionais, como a variável dependente (argumento *y*), para incluir várias análises bivariadas ao relatório.

[4] Versão 0.8.2, disponível em: https://CRAN.R-project.org/package=DataExplorer.

```
# DataExplorer
install.packages("DataExplorer")
library(DataExplorer)

# Nome do arquivo com data e hora para evitar repetição
file.report <- paste("Relatório", format(Sys.time(), "%d-%b-%Y %H:%M"), sep = "
- ")

create_report(
  diamonds,
  output_file = file.report,
  report_title = "Relatório Análise Exploratória",
  y = "price"
)
```

A partir da execução do trecho de código acima, é criado um arquivo em formato HTML no diretório de trabalho, com as informações do relatório, que pode ser aberto em qualquer navegador (Figura 1). O relatório é bastante abrangente, cobrindo grande parte das análises preliminares realizadas durante a AED, apresentando estatísticas básicas e diversos gráficos.

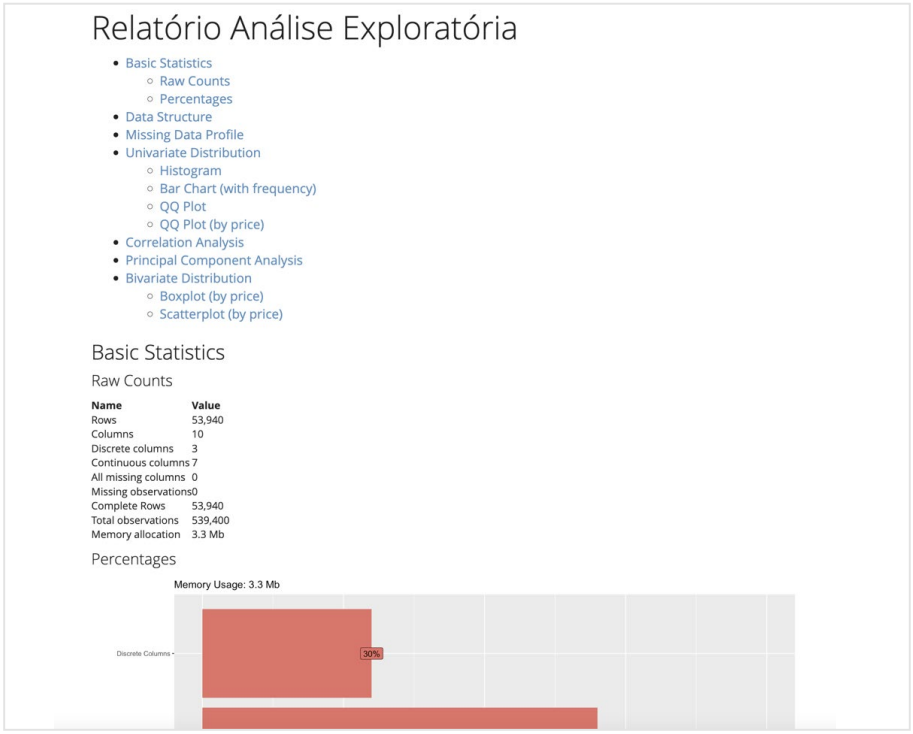


Figura 1. Relatório criado com a função *create_report* do pacote *{DataExplorer}*.

Caso seja desejado ou necessário, análises específicas podem ser realizadas independentemente da geração do relatório, inclusive com a customização das informações e gráficos fornecidos.

Por exemplo, um gráfico de barras da distribuição de frequência (Figura 2) das variáveis categóricas pode ser obtido simplesmente com a execução de apenas um comando:

```
plot_bar (diamonds)
```

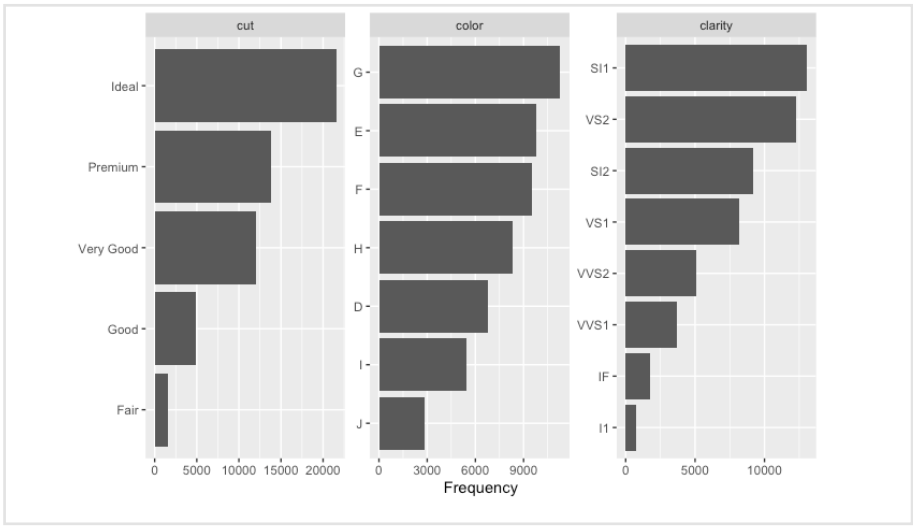


Figura 2. Exemplo de gráfico obtido com a *função plot*_bar do pacote *{DataExplorer}*, apresentando a distribuição de frequência das variáveis categóricas.

O gráfico da Figura 2 é automaticamente gerado com o relatório criado pela função *create_report* (exemplificado acima, na Figura 1). Sendo assim, a partir deste ponto serão apresentados exemplos que envolvem customizações mais detalhadas, com objetivo de explorar as opções disponíveis e os resultados possíveis.

Um gráfico da distribuição de frequências de todas as variáveis categóricas (cor e claridade do diamante) agrupadas pela qualidade do corte (Figura 3), em um tema diferente do padrão (*theme_gray*), de uma forma que seja possível comparar as proporções entre elas, pode ser obtido com:

```
plot_bar(
  diamonds,
  by = "cut",
  ggtheme = theme_light(),
)
```

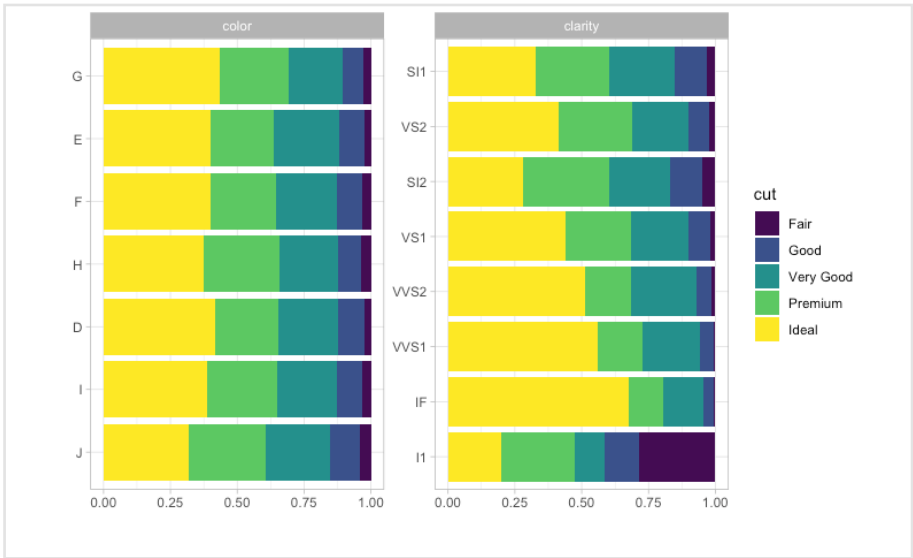


Figura 3. Exemplo de gráfico obtido com a função *plot_bar* do pacote *{DataExplorer}*, permitindo identificar a proporção de diamantes com base na cor, claridade por corte.

Pode-se criar um gráfico que apresente as informações acima de outra maneira: permitindo avaliar quais características determinam os maiores preços (Figura 4). Para isso, colocamos as barras umas ao lado das outras, para facilitar a comparação de valores individuais:

```
plot_bar(
  diamonds,
  by = "cut",
  by_position = "dodge",
  with = "price",
  ggtheme = theme_light(),
  theme_config = list(legend.position = c("bottom"))
)
```

Cabe aqui destacar que, como o *{ggplot2}* permite criar gráficos complexos através de uma interface programática com a qual é possível especificar quais variáveis plotar, como elas serão exibidas e quais propriedades visuais utilizar, pequenas alterações no código criado possibilitam alterar o tipo de

gráfico, usar outros dados ou criar visualizações com elementos adicionais. Esse mesmo conceito se aplica aos gráficos criados com o *{DataExplorer}* e outros pacotes que tenham como base o *{ggplot2}*. Isso é extremamente útil para a criação de gráficos com qualidade para publicações a partir do reuso de códigos criados anteriormente.

Figura 4. Exemplo de gráfico obtido com a função *plot_bar* do pacote *{DataExplorer}*, onde se pode avaliar a influência de cor, claridade e corte sobre o preço de diamantes.

As opções de customização variam de acordo com o tipo de gráfico e estão relacionadas às funções e aos temas associados ao pacote *{ggplot2}* e a outros pacotes que o complementem ou modificam, embora existam opções comuns a todos os tipos de gráficos. Isso permite um grande controle de diversos componentes dos gráficos, conforme o próximo exemplo (Figura 5).

```
plot_histogram(
  diamonds,
  geom_histogram_args = list(
    color = "black",
    fill = "aquamarine3",
    alpha = 0.7,
    bins = 20),
  scale_x = "continuous",
  ggtheme = theme_light(),
  theme_config = list(
    axis.text.x = element_text(
      color = "grey50",
      size = 16,
      angle = 90,
      hjust = 0.5,
      vjust = 0.5),
    axis.text.y = element_text(
      color = "grey50",
      size = 16),
    strip.text = element_text(face = "italic", color = "aquamarine"),
    strip.background = element_rect(fill = "grey40"),
    text = element_text(size = 20)
  )
)
```

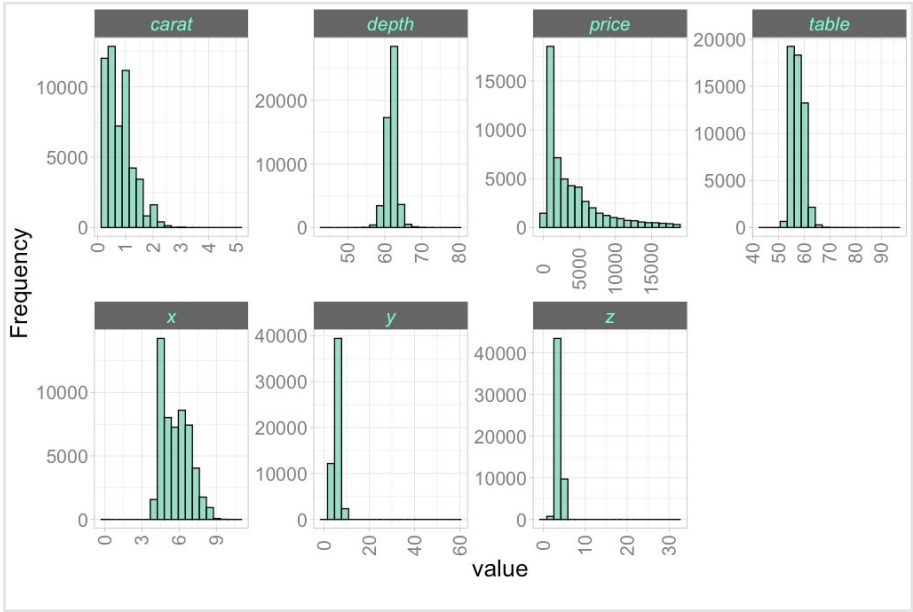


Figura 5. Exemplo de histogramas obtidos com a função *plot_histogram* do pacote *{DataExplorer}*, incluindo diversas customizações.

Outro gráfico bastante útil é o *heatmap* da correlação entre as variáveis (Figura 6). Embora seja possível obter um gráfico de correlação das variáveis categóricas também, através da conversão de seus níveis em variáveis *dummy*, esse gráfico faz parte do relatório. No *snippet* abaixo é gerado um gráfico apenas com as variáveis contínuas.

```
plot_correlation(
  diamonds,
  type = "continuous",
  cor_args = list(method = "pearson"),
  geom_text_args = list(size = 4, color = "navyblue"),
  ggtheme = theme_light(),
  theme_config = list(
    legend.position = "none",
    axis.text.x = element_text(
      size = 16,
      angle = 90,
      color = "grey40",
      hjust = 0.5),
    axis.text.y = element_text(size = 16, color = "grey40"),
    text = element_text(size = 20)
  )
)
```

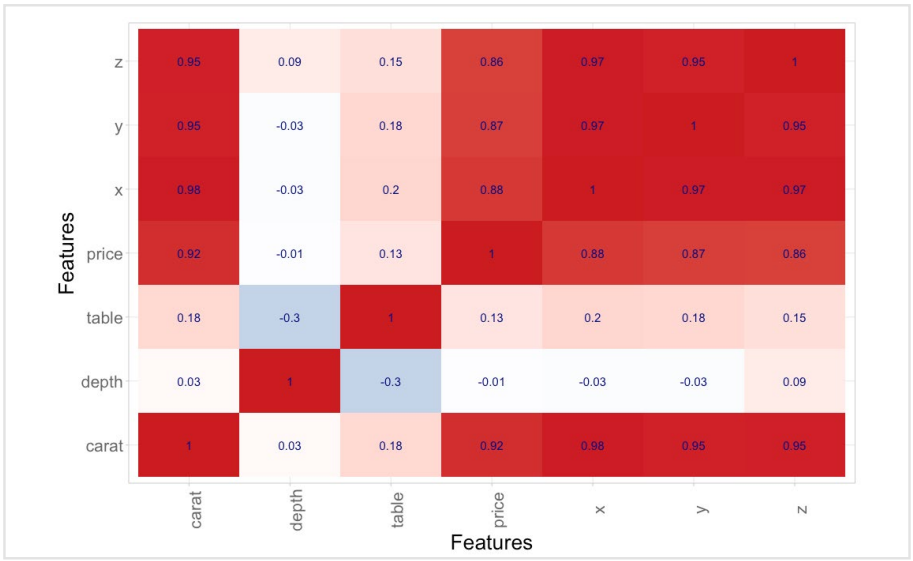


Figura 6. Exemplo de gráfico de correlações obtido com a função *plot_correlation* do pacote *{DataExplorer}*, apenas para as variáveis contínuas do conjunto de dados.

O último exemplo do pacote *{DataExplorer}* é um gráfico quantil-quantil, ou *qq-plot*, das variáveis contínuas principais (Figura 7). Esse tipo de gráfico é utilizado para que se possa verificar a validade de um pressuposto de distribuição de um conjunto de dados:

```
plot_qq(
  data = data.frame(diamonds[, c("carat", "depth", "price", "table", "cut")])
  by = "cut",
  geom_qq_args = list(alpha = 0.2),
  ggtheme = theme_light(),
  theme_config = list(
    axis.text.x = element_text(
      color = "grey50",
      size = 16,
      angle = 90,
      hjust = 0.5,
      vjust = 0.5),
    axis.text.y = element_text(
      color = "grey50",
      size = 16),
    strip.text = element_text(face = "italic", color = "navyblue"),
    strip.background = element_rect(fill = "grey80"),
    text = element_text(size = 20),
    legend.position = c("bottom")
  ),
  nrow = 2L,
  ncol = 2L
)
```

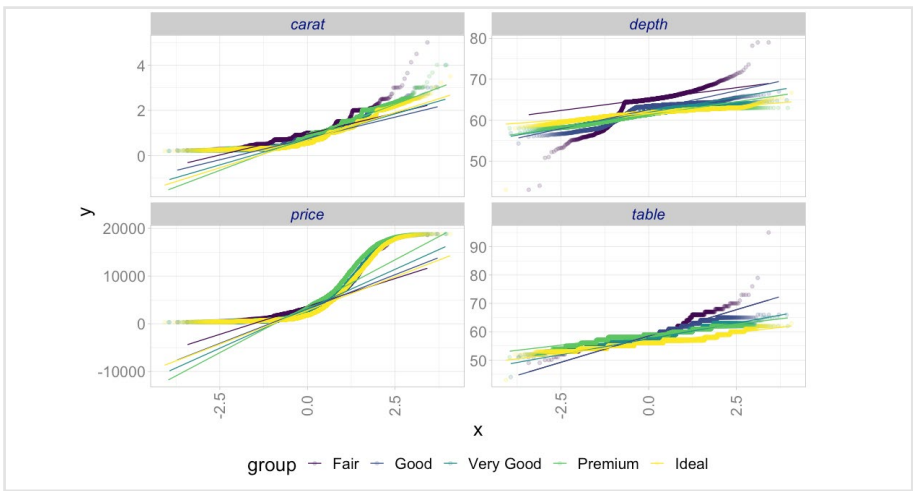


Figura 7. Exemplo de gráfico quantil-quantil obtido com a função *plot_qq* do pacote *{DataExplorer}*, para as variáveis contínuas principais.

O *{DataExplorer}* é uma excelente ferramenta para automatizar a análise exploratória de dados, além de permitir a construção de visualizações complexas e altamente customizáveis devido à sua base ser o *{ggplot2}* e, ainda, possuir diversas funções relativas à engenharia de atributos.

O pacote *{GGally}*

O *{GGally}* (Schloerke et al., 2021) é um pacote de extensão para o *{ggplot2}* que adiciona funcionalidades para visualizar conjuntos de dados de forma automática, reduzindo a complexidade de criar visualizações a partir de transformações nos dados.

Inicialmente, deve-se instalar e carregar o pacote:

```
# GGally
install.packages("GGally")
library(GGally)
```

Como o *{GGally}* é uma extensão para o *{ggplot2}*, a forma de ser executado segue o conceito de camadas e todas as opções e funcionalidades do *{ggplot2}* podem ser utilizadas.

Uma matriz de gráficos (Figura 8) com variáveis pareadas (*generalized pairs plot;* Emerson et al., 2012), considerando apenas algumas variáveis do conjunto de dados (por uma questão didática), pode ser obtida, por exemplo, com:

```
pairs.plot <- ggpairs(
  diamonds,
  columns = c("carat", "depth", "price", "table"),
  lower = list(
    continuous = "smooth",
    combo = "facetdensity",
    mapping = aes(color = cut)
  ),
  legend = c(2,1)
)

# Opções estéticas
pairs.plot + theme_light() + theme(legend.position = "bottom")
```

Desta vez gráfico resultante foi atribuído a uma variável (*pairs.plot*), para permitir que ajustes posteriores, como o de tema e legenda, fossem realizados, seguindo a proposta e o padrão da construção em camadas do *{ggplot2}*.

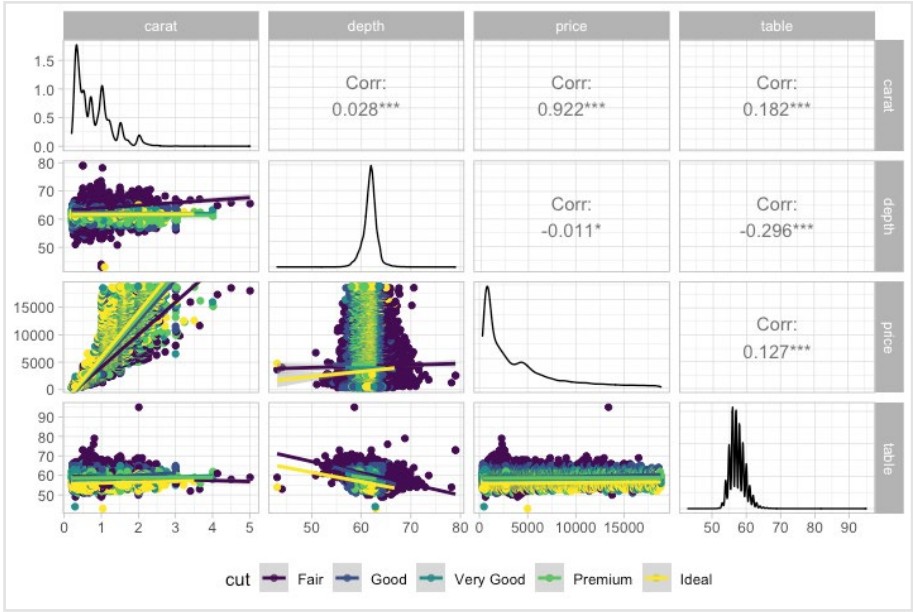


Figura 8. Exemplo de matriz de gráficos obtida com a função *ggpairs* do pacote *{GGally}*, para as variáveis contínuas principais do conjunto de dados.

Uma matriz de gráficos de duas variáveis agrupadas (Figura 9) pode ser feita com a função *ggduo*. Isso é útil para análise de correlação canônica, análise de várias séries temporais e análise de regressão. Um exemplo desse gráfico, pode ser conseguido com o *snippet* abaixo, que inclui algumas modificações na apresentação básica através de configurações no tema:

```
duo.plot <- ggduo(
  diamonds,
  c("clarity", "color"),
  c("price"),
  mapping = aes(color = cut),
  types = list(continuous = "smooth_lm", comboVertical = "box_no_facet"),
  legend = c(1,1)
)

# Opções estéticas
duo.plot + theme_light() + theme(
  legend.position = "bottom",
  strip.text = element_text(face = "italic", color = "navyblue", size = 20),
  strip.background = element_rect(fill = "grey80"),
  axis.text = element_text(size = 16)
)
```

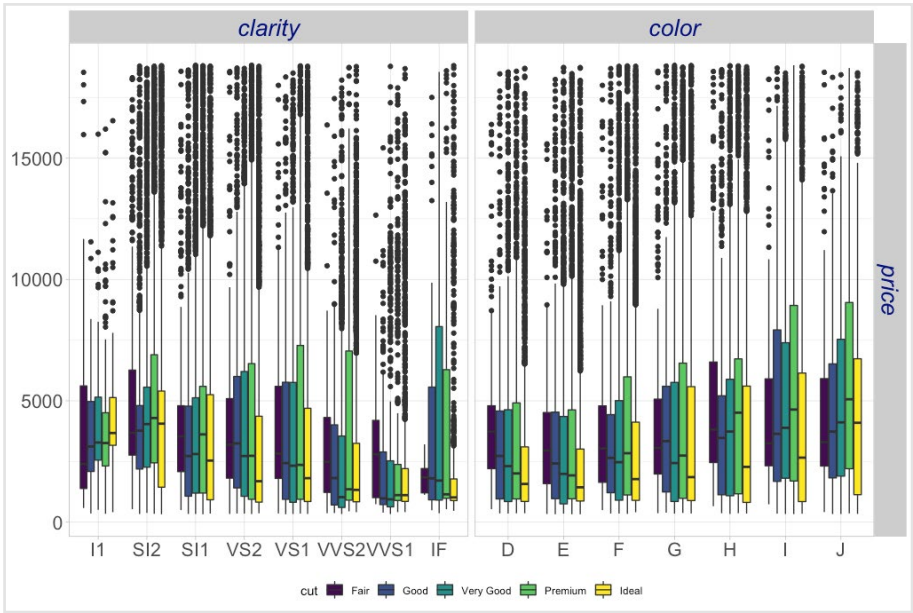


Figura 9. Exemplo de matriz de gráficos obtida com a função *ggduo* do pacote *{GGally}*.

Outro gráfico extremamente útil é relativo ao diagnóstico de modelos, com as informações referentes às variáveis explanatórias utilizadas. A visualização do resultado do ajuste de um modelo é conseguida com a função *ggnostic*. Abaixo, um exemplo onde, inicialmente, um modelo linear é ajustado (*price.model*):

```
price.model <- step(lm(price ~ carat + cut + table, data = diamonds))

diag.plot <- ggnostic(price.model, mapping = aes(color = cut))

# Opções estéticas
diag.plot + theme_light() + theme(
  legend.position = "bottom",
  strip.text = element_text(
    face = "italic",
    color = "navyblue",
    size = 16
  ),
  strip.background = element_rect(fill = "grey80"),
  axis.text = element_text(size = 12),
  axis.text.x = element_text(
    size = 10,
    angle = 90,
    hjust = 0.5,
    vjust = 0.5
  )
)
```

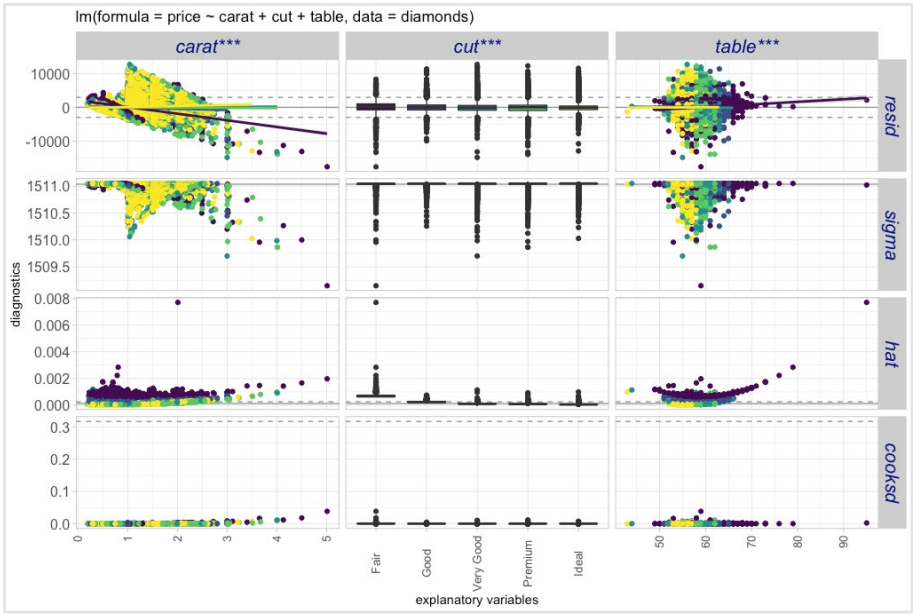


Figura 10. Exemplo de gráfico de diagnóstico de ajuste de modelos com a função *ggnostic* do pacote *{GGally}*.

Tabelas de contingência podem ser obtidas de diversas formas através do uso da função *ggtable*. Os valores podem ser absolutos, proporcionais ao total ou às linhas e colunas. As células podem ser coloridas de acordo com seus resíduos padronizados indicando se estão acima ou abaixo das frequências esperadas, de acordo com Chi-quadrado. O *snippet* abaixo ilustra uma tabela que pode ser obtida (Figura 11):

```
table <- ggtable(
  diamonds,
  c("cut"),
  c("color", "clarity"),
  cells = "row.prop"
)

# Opções estéticas
table + theme_light() + theme(
  legend.position = "bottom",
  strip.text = element_text(
    face = "italic",
    color = "navyblue",
    size = 20),
  strip.background = element_rect(fill = "grey80"),
  axis.text = element_text(size = 16),
)
```

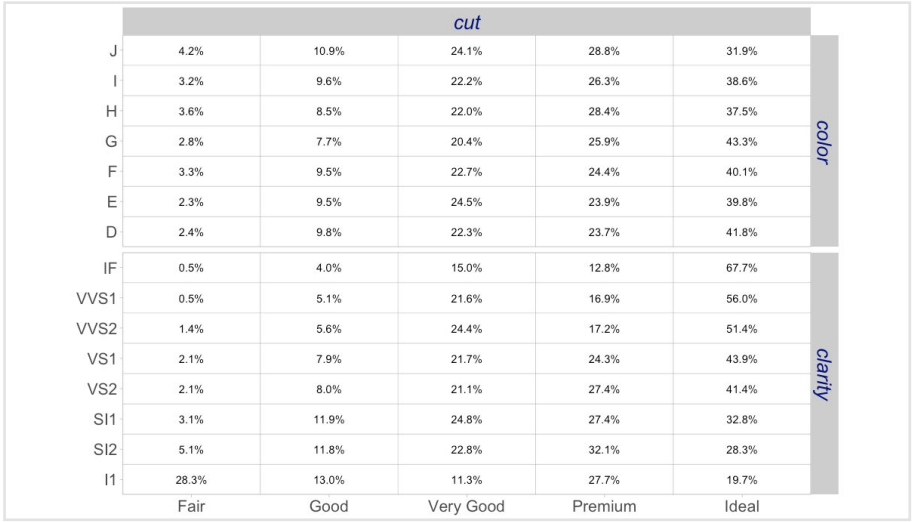


Figura 11. Exemplo de tabela de contingência obtida com a função *ggnostic* do pacote *{GGally}*, com valores proporcionais relativos às linhas.

O pacote possui ainda várias outras funções adicionais para construção de gráficos, como redes sobre mapas, curvas de sobrevivência, *glyphplots*. Nos exemplos apresentados fica clara sua simplicidade e versatilidade, especialmente por possibilitar customizações com os recursos do *{ggplot2}*.

O pacote *{SmartEDA}*

O pacote *{SmartEDA}* (Putatunda et al., 2019) seleciona automaticamente as variáveis e realiza estatísticas descritivas relacionadas. Além disso, também permite a obtenção de tabelas personalizadas, estatísticas resumidas e se utiliza de diversas técnicas de visualização para auxiliar a avaliação de variáveis numéricas e categóricas.

De forma semelhante ao *{DataExplorer}*, um relatório completo pode ser obtido a partir da execução, por exemplo, do código abaixo:

```
# SmartEDA
install.packages("SmartEDA")
library(SmartEDA)

ExpReport(
  diamonds,
  Target = "price",
  label = NULL,
  op_file = "Relatório EDA.html"
)
```

O arquivo “Relatório EDA.html” será criado no diretório de trabalho atual, com o resultado da análise exploratório padrão do pacote (Figura 12), abaixo:

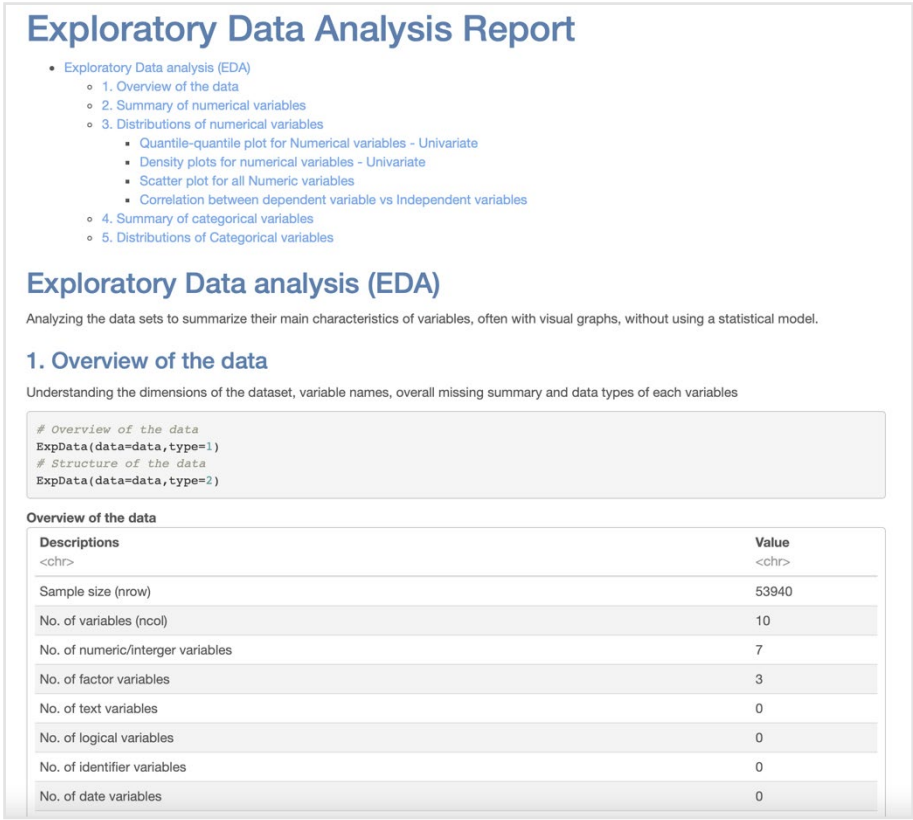


Figura 12. Relatório criado com a função *ExpReport* do pacote *{SmartEDA}*.

Além do relatório, é possível obter diversas tabelas descritivas que podem ser formatadas para utilização diretamente em documentos Word (o pacote *{flextable}* é uma opção para isso) e vários gráficos.

Estatísticas descritivas podem ser personalizadas, inclusive com o uso de filtros, e obtidas facilmente, com, por exemplo:

```
ExpCustomStat(
  diamonds,
  Nvar = c("price","carat","table","depth"),
  Cvar = c("cut"),
  stat = c("Count", "Prop", "mean", "sd", "var", "min", "max")
)
```

Que resulta na descrição dos dados da seguinte forma:

```
          cut Attribute Count  Prop         mean           sd              var    min      max
 1:     Ideal     price 21551 39.95 3457.5419702 3808.4011723 14503919.4895493 326.00 18806.00
 2:   Premium     price 13791 25.57 4584.2577043 4349.2049615 18915583.7971080 326.00 18823.00
 3:      Good     price  4906  9.10 3928.8644517 3681.5895839 13554101.8643956 327.00 18788.00
 4: Very Good     price 12082 22.40 3981.7598907 3935.8621606 15491010.9469782 336.00 18818.00
 5:      Fair     price  1610  2.98 4358.7577640 3560.3866123 12676352.8287930 337.00 18574.00
 6:     Ideal     carat 21551 39.95    0.7028370    0.4328763        0.1873819   0.20     3.50
 7:   Premium     carat 13791 25.57    0.8919549    0.5152616        0.2654945   0.20     4.01
 8:      Good     carat  4906  9.10    0.8491847    0.4540544        0.2061654   0.23     3.01
 9: Very Good     carat 12082 22.40    0.8063814    0.4594354        0.2110809   0.20     4.00
10:      Fair     carat  1610  2.98    1.0461366    0.5164043        0.2666734   0.22     5.01
11:     Ideal     table 21551 39.95   55.9516681    1.2464233        1.5535711  43.00    63.00
12:   Premium     table 13791 25.57   58.7460953    1.4785733        2.1861789  51.00    62.00
13:      Good     table  4906  9.10   58.6946392    2.8512997        8.1299101  51.00    66.00
14: Very Good     table 12082 22.40   57.9561496    2.1214481        4.5005420  44.00    66.00
15:      Fair     table  1610  2.98   59.0537888    3.9462613       15.5729782  49.00    95.00
16:     Ideal     depth 21551 39.95   61.7094010    0.7185386        0.5162977  43.00    66.70
17:   Premium     depth 13791 25.57   61.2646726    1.1588149        1.3428520  58.00    63.00
18:      Good     depth  4906  9.10   62.3658785    2.1693739        4.7061831  54.30    67.00
19: Very Good     depth 12082 22.40   61.8182751    1.3786308        1.9006229  56.80    64.90
20:      Fair     depth  1610  2.98   64.0416770    3.6434275       13.2745640  43.00    79.00
```

Gráficos mais complexos podem ser facilmente obtidos como, por exemplo, dois gráficos independentes, lado a lado, para a mesma variável:

```
ExpTwoPlots(
  data = diamonds,
  plot_type = "numeric",
  iv_variables = c("price", "carat"),
  target = "cut",
  lp_arg_list = list(binwidth = 1),
  lp_geom_type = 'qqplot',
  rp_arg_list = list(alpha = 0.4, binwidth = 1),
  rp_geom_type = 'density',
  fname = "dub",
  page = c(2,1)
)
```

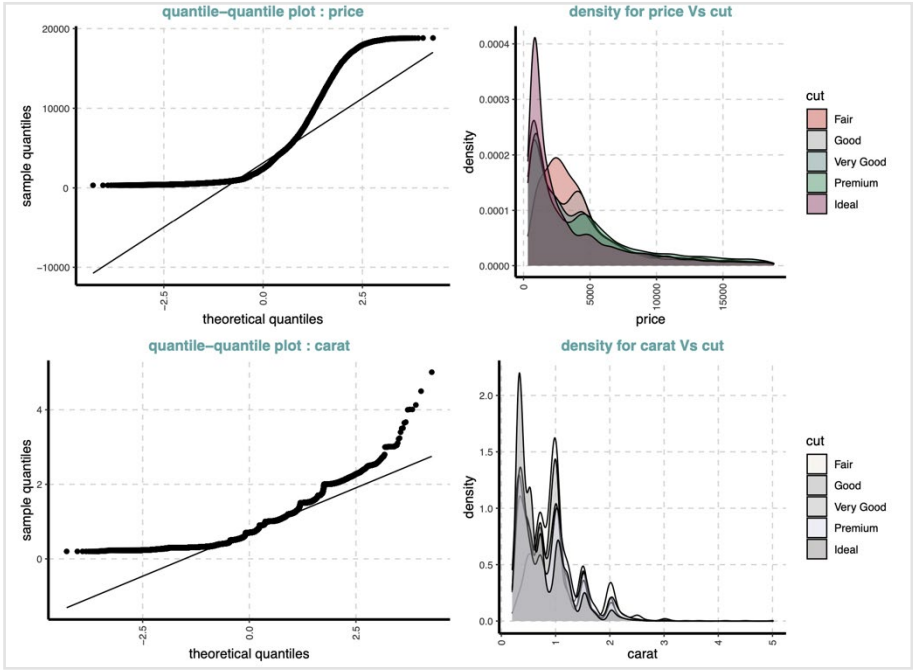


Figura 13. Exemplo de gráfico duplo criado com a função *ExpTwoPlots* do pacote *{SmartEDA}*, para algumas variáveis contínuas do conjunto de dados.

Um gráfico de coordenadas paralelas (Figura 14), que é utilizado para comparar as relações existentes entre variáveis que possuem magnitudes ou tipos diferentes em uma única visualização, é facilmente obtido através do uso da função *ExpParcoord*:

```
ExpParcoord(
  data = diamonds,
  Nvar = c("price", "depth", "carat", "table"),
  Group = "cut",
  scale = "uniminmax"
)
```

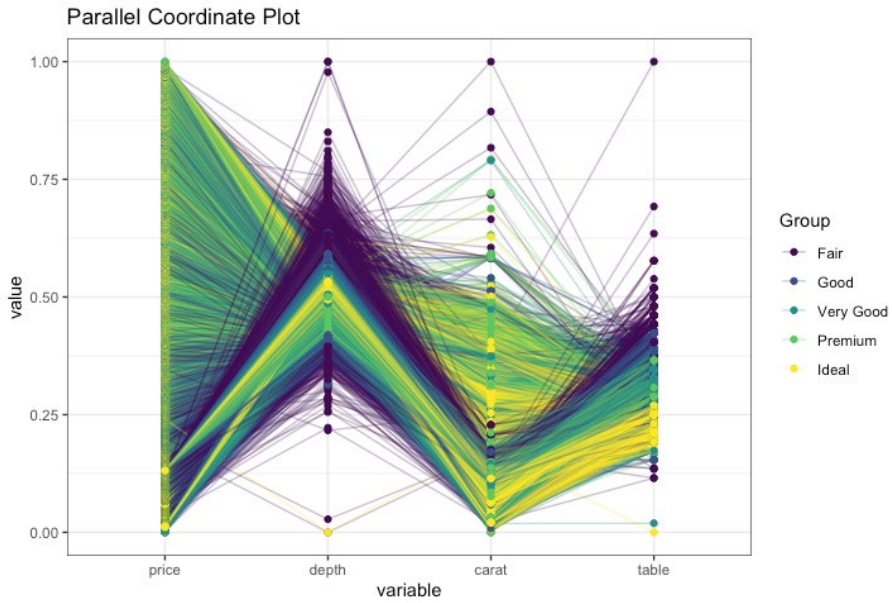


Figura 14. Exemplo de gráfico duplo criado com a função *ExpParcoord* do pacote *{SmartEDA}*, para algumas variáveis contínuas do conjunto de dados.

Diversas opções estão disponíveis para alterar vários aspectos dos gráficos que podem ser criados, possibilitando análises em vários níveis e com diferentes formas de combinar as variáveis.

Considerações Finais

Neste trabalho foram apresentados alguns dos pacotes disponíveis para o ambiente R para executar, de forma automática, a maioria das atividades relacionadas à análise exploratória de dados, desde a criação de tabelas de estatísticas descritivas das variáveis envolvidas, passando por tabelas de contingência e tabelas para publicação, visualização do conjunto de dados em diversos aspectos e incluindo a criação de relatórios completos.

Foram utilizados exemplos de códigos simples e mais elaborados, abordando diferentes aspectos e algumas das opções disponíveis para customização cosmética dos resultados, criando gráficos e tabelas mais adequados para

visualizar as relações existentes entre as variáveis, suas distribuições, existência de *outliers* e outros aspectos.

As ferramentas apresentadas permitem contar uma história adequada com os dados, ou seja, compartilhar as respostas encontradas a partir das análises realizadas, para diferentes audiências, a partir da seleção, dentre as diversas opções e resultados possíveis, das informações que sejam mais relevantes para isso.

Essencialmente, contar uma história a partir de dados, de forma efetiva, envolve descrever brevemente os conjuntos de dados utilizados, explicar claramente por que esses dados estão sendo analisados e resumir de forma objetiva o que foi aprendido ao analisá-los. Sendo assim, tão importante quanto uma boa análise exploratória, é conhecer a audiência para a qual se destina o resultado dessa análise e, nesse contexto, possuir diferentes ferramentas que possibilitam avaliar diferentes aspectos dos dados, é fundamental. Isso foi apresentado neste trabalho.

Referências

DIACONIS, P. **Theories of Data Analysis**: From Magical Thinking Through Classical Statistics. In: EXPLORING data tables, trends, and shapes. [S.l.]: Wiley, 2006. p. 1-36. (Wiley Series in Probalility and Statistics).

EMERSON, J. W.; GREEN, W. A.; SCHLOERKE, B.; CROWLEY, J.; COOK, D.; HOFMANN, H.; WICKHAM, H. The Generalized Pairs Plot. **Journal of Computational and Graphical Statistics**, v. 22, n. 1, p. 79-91, 2012.

HUNTINGTON-KLEIN, N. **vtable:** Variable Table for Variable Documentation. [s.l: s.n.], 2022.

ILIINSKY, N.; STEELE, J. **Designing data visualizations:** Representing informational Relationships. [S.l.]: O'Reilly Media, 2011.

PEARSON, R. K. **Exploratory Data Analysis Using R**. Chapman and Hall/CRC, 2018. 563 p.

PUTATUNDA, S.; UBRANGALA, D.; KIRAN, D.; KONDAPALLI, R. SmartEDA: An R Package for Automated Exploratory Data Analysis. J**ournal of Open Source Software**, v. 4, n. 41, p. 1509, 2019.

R CORE TEAM. **R:** A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing, Disponível em: https://www.R-project.org/. Acesso em: 26 jul. 2022.

SCHLOERKE, B.; COOK , D.; LARMARANGE, J.; BRIATTE, F.; MARBACH, M.; THOEN, E.; ELBERG, A.; TOOMET, O.; CROWLEY, J.; HOFMANN, H.; WICKHAM, H. **GGally:** Extension to “ggplot2”. [S.l.: s.n.], 2021.

SJOBERG, D. D.; WHITING, K.; CURRY, M.; LAVERY, J. A.; LARMARANGE, J. Reproducible Summary Tables with the gtsummary Package. **The R Journal**, v. 13, n. 1, p. 570–580, 2021.

WICKHAM, H. **ggplot2**: Elegant Graphics for Data Analysis. [s.l.] New York: Springer-Verlag, 2016.

YOSHIDA, K.; BARTEL, A. **tableone:** Create ‘Table 1’ to Describe Baseline Characteristics with or without Propensity Score Weights. R package version 0.13.2. 2022. Disponível em: https://CRAN.R-project.org/package=tableone. Acesso em: 14 mar. 2023.

Embrapa
Roraima

MINISTÉRIO DA
AGRICULTURA E
PECUÁRIA

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO