SME0820 - Modelos de Regressão e Aprendizado Supervisionado I - Trabalho I

Brenda da Silva Muniz 11811603 Francisco Rosa Dias de Miranda 4402962
Heitor Carvalho Pinheiro 11833351- Mônica Amaral Novelli 11810453

Setembro 2021

Neste trabalho, nosso objetivo é ajustar um modelo de regressão linear simples ao conjunto de dados fornecido, utilizando linguagem R. Para esta tarefa, descreveremos cada etapa de nosso *pipeline*.

Primeiramente, vamos carregar os módulos utilizados nesta análise. Caso não possua algum dos pacotes, utilize o comando `install_packages("Nome_do_pacote")`.

```
library(tidyverse)
library(ggpubr)
library(corrplot)
library(DataExplorer)
library(GGally)
library(knitr)
```

Com os pacotes carregados em nosso ambiente, lemos o arquivo `.csv` disponibilizado colocando-o na mesma pasta de nosso projeto. Vamos inspecionar o que foi carregado com auxílio do comando `str()` para mostrar o tipo de cada uma das colunas de nosso dataset.

```
dados <- read_csv("data-table-B3.csv", locale = locale(decimal_mark = ","))

str(dados)
```

```
## spec_tbl_df[,12] [32 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ y  : num [1:32] 18.9 17 20 18.2 20.1 ...
##  $ x1 : num [1:32] 350 350 250 351 225 440 231 262 89.7 96.9 ...
##  $ x2 : num [1:32] 165 170 105 143 95 215 110 110 70 75 ...
##  $ x3 : num [1:32] 260 275 185 255 170 330 175 200 81 83 ...
##  $ x4 : num [1:32] 8 8.5 8.25 8 8.4 8.2 8 8.5 8.2 9 ...
##  $ x5 : num [1:32] 2.56 2.56 2.73 3 2.76 2.88 2.56 2.56 3.9 4.3 ...
##  $ x6 : num [1:32] 4 4 1 2 1 4 2 2 2 2 ...
##  $ x7 : num [1:32] 3 3 3 3 3 3 3 3 4 5 ...
##  $ x8 : num [1:32] 200 200 197 200 194 ...
##  $ x9 : num [1:32] 69.9 72.9 72.2 74 71.8 69 65.4 65.4 64 65 ...
##  $ x10: num [1:32] 3910 3860 3510 3890 3365 ...
##  $ x11: num [1:32] 1 1 1 1 0 1 1 1 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   y = col_double(),
##   ..   x1 = col_double(),
```

```
##   ..   x2 = col_double(),
##   ..   x3 = col_double(),
##   ..   x4 = col_double(),
##   ..   x5 = col_double(),
##   ..   x6 = col_double(),
##   ..   x7 = col_double(),
##   ..   x8 = col_double(),
##   ..   x9 = col_double(),
##   ..   x10 = col_double(),
##   ..   x11 = col_double()
##   .. )
```

Parte a):

- Descrição do banco de dados

Poderíamos também descrever as colunas do banco de dados com auxílio da função `introduce()` do pacote `DataExplorer`. A tabela obtida é exibida abaixo:

```
a <- introduce(dados)


a %>% select(rows,
             discrete_columns,
             continuous_columns,
             total_observations,
             complete_rows,
             total_missing_values) %>%
  kable(
    col.names = c("Linhas", "Colunas Discretas",
                  "Colunas Contínuas", "Total de observações",
                  "Atributos sem NA", "Atributos com NA"))
```

Linhas	Colunas Discretas	Colunas Contínuas	Total de observações	Atributos sem NA	Atributos com NA
32	0	12	384	30	2

- Definição das variáveis
- Análise exploratória inicial

```
ggcorr(dados, geom = "circle")
```

- Graficos de dispersão Y versus X_i, $i = 1, ..., 11$.

```
dados %>%
  pivot_longer(cols = !"y") %>% #todas as variaveis como funcao de y
  ggplot(aes(y = y)) +
    geom_point(aes(x = value)) +
    facet_wrap(~name, scales = "free_x") + theme_pubclean()
```

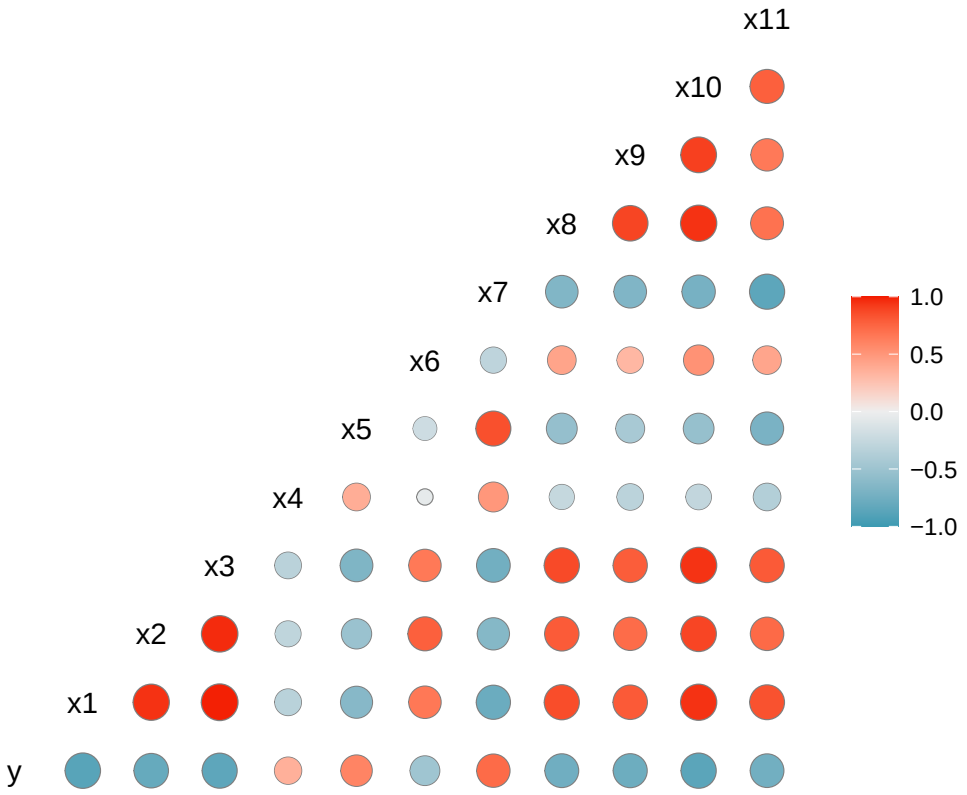
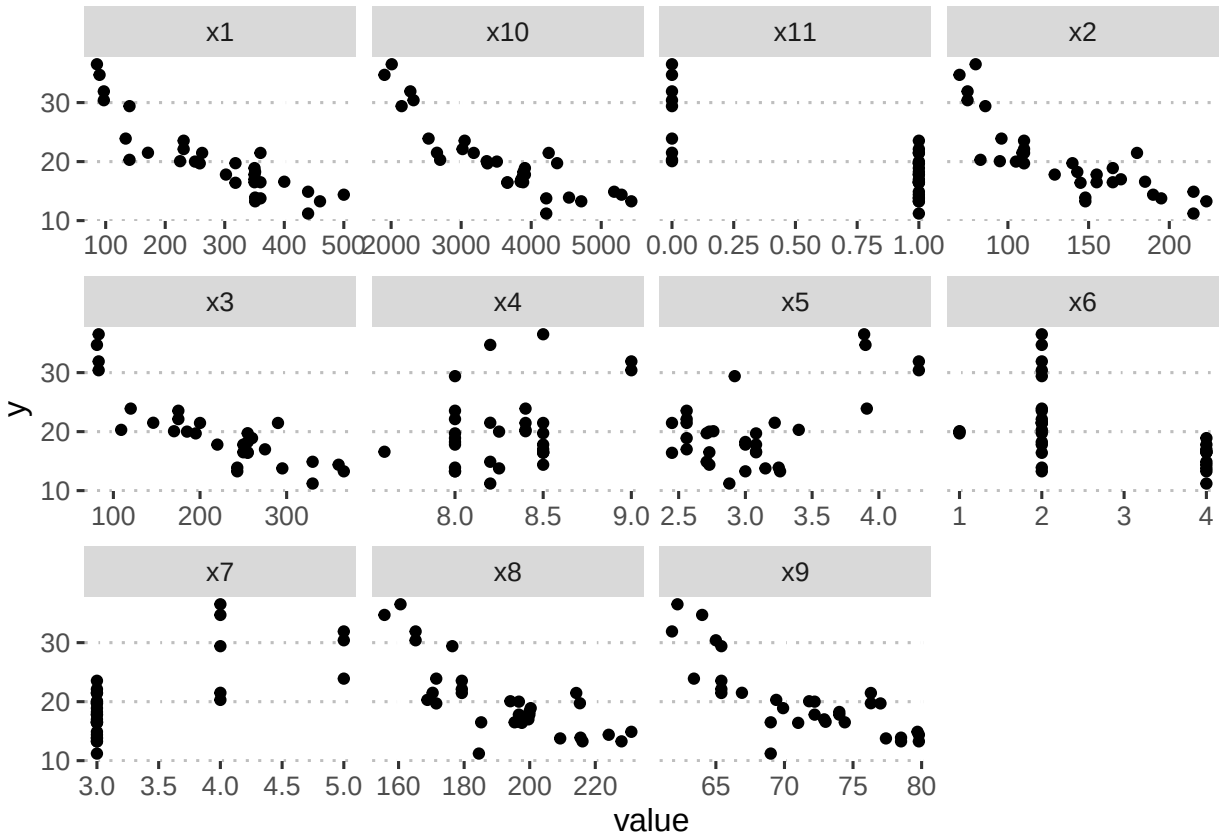


Figure 1: Correlograma entre as variáveis



Interpretação de cada gráfico

Parte b):

Consultar e descrever brevemente os conceitos Data splitting, cross validation, overfitting, underfitting, missing data, encoding data.

Parte c):

1. Calcular S_{XX},S_{YY} e S_{XY}
2. Ajustar um modelo de regressão linear simples, apresentar a estimativa de β_0, β_1 e σ^2 e fazer um gráfico com a reta ajustada
3. Calcule o valor dos $\hat{Y}$ e o valor dos resíduos para seu modelo, resumo e histograma dos resíduos, e faça uma análise da distribuição destes.
4. testes de hipotese para β_0 e β_1
5. intervalos de confiança
6. intervalos de predição
7. tabela anova