

# Análise de Regressão Linear

## Grupo 4

Heitor Augustaitis de Oliveira  
Lucas Moura de Carvalho  
Bruno Sergio Procopio Junior  
Thales Simão do Amaral Camargo  
Matheus Taipina Benini  
Bruno de Oliveira Feitosa

28/09/2020

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Motivação . . . . .	3
1.2	Preparação do ambiente . . . . .	3
<b>2</b>	<b>Análise Inicial</b>	<b>4</b>
2.1	Correlações . . . . .	4
2.2	Tratamento de outliers . . . . .	5
<b>3</b>	<b>Análise de Regressão</b>	<b>7</b>
<b>4</b>	<b>Referências</b>	<b>9</b>

# 1 Introdução

## 1.1 Motivação

Este relatório tem como objetivo mostrar as análises feitas pelo grupo, utilizando a linguagem R e a técnica de regressão linear, sobre a base de dados *summer-products-with-rating-and-performance\_2020-08*. A preparação do ambiente de trabalho e as ferramentas empregadas serão apresentadas, assim como os comandos utilizados para processar os dados e suas respectivas saídas.

Uma questão natural é se os produtos seguem a economia de escala, onde quanto mais se vende mais se diminui o preço dele. A relação entre as unidades vendidas e a sua nota geral pode ser usada como teste de sanidade da suposição de que quanto mais se vende mais avaliado é um item. Outra questão interessante é o comportamento do preço do varejo (do consumidor final) em relação ao preço. Por fim as colunas em relação ao vendedor foram selecionadas para adicionar variedade a análise, vendedores com notas maiores tem maior nota de produtos, menores preços?

## 1.2 Preparação do ambiente

Para as nossas análises, utilizamos a linguagem R e as ferramentas R Studio e Jupyter Notebook. Tanto o script em R (.r) quanto o notebook (.ipynb) utilizado estão disponíveis aqui. Duas bibliotecas além do “base” foram utilizadas, a **dplyr** que é uma coleção de ferramentas para facilitar o manuseio de objetos como *data frames* e o **ggplot2** que é excelente para a criação de gráficos.

Execução dos comandos de carregamento da base dados e seleção dos campos a serem trabalhados:

```
# Le o dataset com as colunas desejadas
dataset <- read.csv(file="./summer-products-with-rating-and-performance_2020-08.csv", header=TRUE, sep=",")
dataset <- as.data.frame(dataset)
base <- select(dataset, price, units_sold, rating, rating_count, retail_price, merchant_rating, merchant_rating_count)

# Nomes alternativos
colnames(base) <- c("Price", "Units Sold", "Rating", "Rating C", "Retail Price", "Merch R", "Merch C")
```

A base de dados veio sobre formato *.csv* e sobre ela escolhemos sete colunas que condizem com as motivações da escolha da base, eles foram renomeadas de acordo com a tabela abaixo.

Coluna original	Coluna Nova	Significado
price	Price	preço do item
units_sold	Units Sold	número de unidades vendidas
rating	Rating	Sumário das notas dadas ao produto
rating_count	Rating C	quantidade de notas dadas ao produto
retail_price	Retail Price	preço de varejo
merchant_rating	Merch R	sumário das notas dadas ao vendedor
merchant_rating_count	March C	quantidade das notas dadas ao vendedor

*Tabela de colunas*

## 2 Análise Inicial

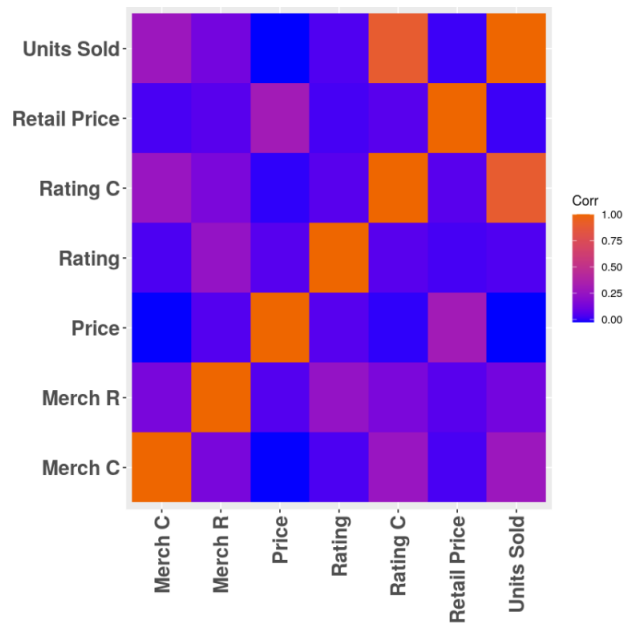
### 2.1 Correlações

A primeira análise foi feita com uma matriz de correlação das colunas, a baixo está o seu mapa de calor.

```
# Mapa de calor de correlacao
cor_map = data.frame(rows = rep(colnames(base), each = ncol(base)), cols = rep(colnames(base), each = 1, times=ncol(base)), Corr = c(cor(base)), stringsAsFactors=FALSE)

cor_heat = ggplot(cor_map, aes(rows, cols)) + geom_tile(aes(fill = Corr))
cor_heat = cor_heat + scale_fill_gradient(low = "#0000FF", high = "#EE6600") + theme(axis.title.y=element_blank(), axis.title.x=element_blank(), axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1, size=16, face="bold"), axis.text.y = element_text(vjust = 0.5, hjust=1, size=16, face="bold"))
cor_heat

# Eliminares os valores com correlacao quase perfeita e muita baixa
cor_map = subset(cor_map, (abs(Corr) < 0.99 & abs(Corr) > 0.2))
```

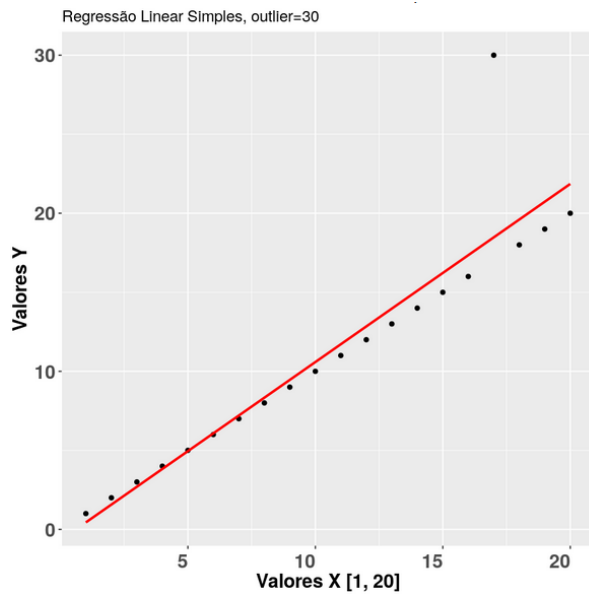


Matriz de Correlação

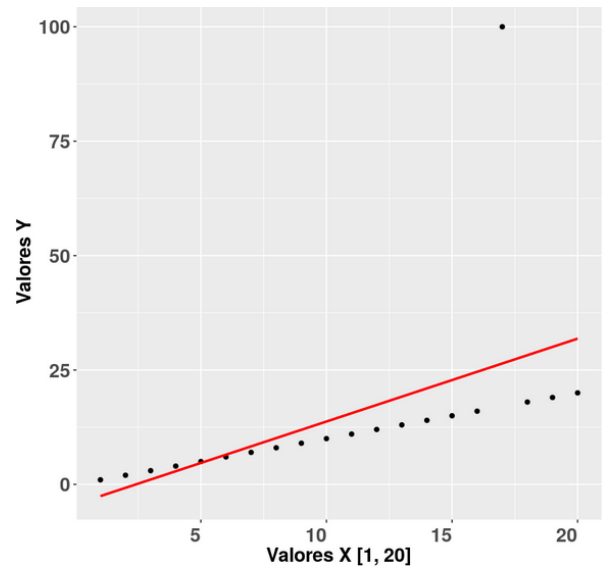
Observando a matriz, nota-se uma correlação com valor alto além das identidades, que é o de Rating C com Units Sold, o que era de se esperar. Dessa matriz foram eliminados os valores com correlação menor que 0.2 já que não há linearidade aparente significativa.

## 2.2 Tratamento de outliers

Para uma regressão linear, a existência de outliers pode ser especialmente nociva uma vez que as posições dos valores é levada em consideração de maneira linear e ascendente. Nas imagens abaixo fica nítido esse problema, onde dos pontos da curva  $f(x) = x$  há apenas um outlier (18, 30) na primeira (1) e (18, 100) na segunda (2), que jogam a reta do modelo para longe dos outros pontos.



1. Regressão com outlier = 30



2. Regressão com outlier = 100

Foi criada uma função que remove linhas do dataset, se baseando em uma das colunas. É calculado o intervalo interquartil e são eliminados os valores além dos limites superior e inferior.

```
## Seção de outliers
# Quartis
Q1 = 0.25
Q2 = 0.5
Q3 = 0.75
Q4 = 1.0

# Calcula Inter Quartile Range (IQR) da lista
iqrang <- function(arr)
{
  quantile(arr, Q3) - quantile(arr, Q1)
}

# Limpa outliers se baseando em uma coluna
limpaOutlier <- function(df, col)
{
  IQR = iqrang(unlist(df[[col]]))
  lower = Q1 - 1.5*IQR
  upper = Q3 + 1.5*IQR
  subset(df, df[[col]] > lower & df[[col]] < upper)
}
```

3. Código para outliers

$$IQQ = Q3 - Q1$$

$$LS = Q3 + 1.5IQQ$$

$$LI = Q1 - 1.5IQQ$$

4. Definições de IQQ, LS, LI

Sobre a matriz de correlação foram aplicados os tratamentos de outliers e foram selecionados apenas os pares que sofreram um aumento na correlação.

```
# Calcula novo cor_map aplicando remocao de outliers 2 a 2

cor_map$Corr_New <- NA
cor_rows = cor_map$"rows"
cor_cols = cor_map$"cols"

for(i in 1:(nrow(cor_map)))
{
  sub_base = limpaOutlier(base, cor_rows[[i]])
  sub_base = limpaOutlier(sub_base, cor_cols[[i]])
  sub_base = select(sub_base, cor_rows[i], cor_cols[[i]])
  cor_map[i, "Corr_New"] = cor(sub_base)[2]
}
cor_map_new <- subset(cor_map, abs(cor_map$Corr_New) > abs(cor_map$Corr))
cor_map
cor_map_new
```

## 5. Tratamento de outliers 2 a 2

	rows	cols	Corr	Corr_New
	<chr>	<chr>	<dbl>	<dbl>
5	Price	Retail Price	0.3047476	0.3992272
14	Units Sold	Merch C	0.2728973	0.2758841
28	Rating C	Merch C	0.2581676	0.2826965

## 6. Resultados

# 3 Análise de Regressão

Baseando-se na alta correlação encontrada entre Rating Count e Units Sold, foi feita a análise de regressão linear para essas duas variáveis, sendo Units Sold a variável dependente e Rating Count a variável independente.

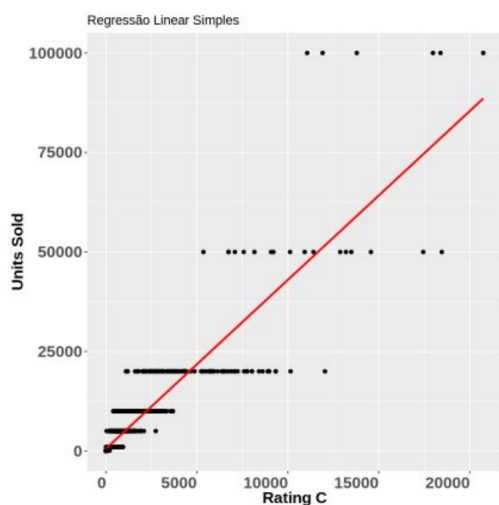
É possível observar alguns resultados positivos da análise, como o valor 0.809 para o  $R^2$  e o baixo valor para p-value, mostrando que a variável possui significância estatística para o modelo.

```
xc = "Rating C"
yc = "Units Sold"

# Normal
lmGraph(base, NULL, base[[xc]], base[[yc]], xcol_name=xc, ycol_name=yc)

# Sem outlier
sub_base = limpaOutlier(base, xc)
sub_base = limpaOutlier(sub_base, yc)
lmGraph(sub_base, NULL, sub_base[[xc]], sub_base[[yc]], xcol_name=xc, ycol_name=yc)
```

A figura a seguir (7) mostra o gráfico de dispersão entre as variáveis Rating Count e Units Sold. Observa-se uma linearidade entre as variáveis, apesar do comportamento estranho da variável Units Sold, que possui os dados agrupados em valores específicos. Isso se deve ao fato de que, provavelmente, os valores dessa coluna foram arredondados para faixas específicas de valores, o que causou a aparência atípica dos dados.



```
Call:
lm(formula = xcol ~ ycol)

Residuals:
    Min       1Q   Median       3Q      Max
-8072.0  -121.8   -61.8    8.7   8864.9

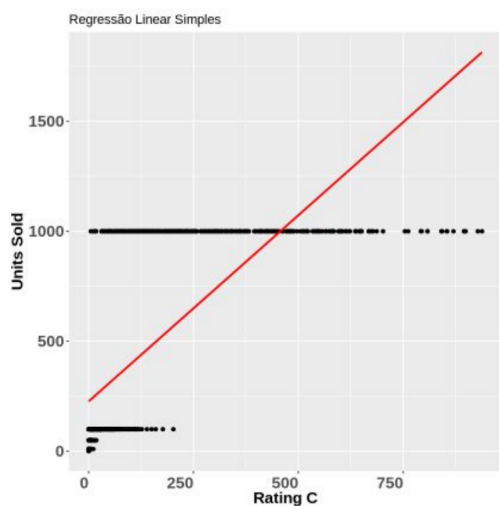
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.127586  24.104570   2.577   0.01 *
ycol         0.190719   0.002338  81.582 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 867.2 on 1571 degrees of freedom
Multiple R-squared:  0.809,    Adjusted R-squared:  0.8089
F-statistic: 6656 on 1 and 1571 DF, p-value: < 2.2e-16
```

## 7. Regressão Rating C e Units Sold

## 8. Sumário

O mesmo procedimento de regressão linear foi executado , porém retirando os outliers das duas variáveis utilizadas. O resultado foi um  $R^2$  muito menor, e um p-value muito alto, o que retira a significância estatística da variável. Isso foi feito pois os valores da parte superior da ordenada são pequenos em número porém algumas vezes maiores em magnitude que os valores na parte inferior.



```
Call:
lm(formula = xcol ~ ycol)

Residuals:
    Min       1Q   Median       3Q      Max
-275.38  -28.93   -9.93    20.07   657.62

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.216766   5.677553   0.567   0.571
ycol          0.277159   0.009096  30.472 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 131.5 on 1051 degrees of freedom
Multiple R-squared:  0.4691,    Adjusted R-squared:  0.4686
F-statistic: 928.5 on 1 and 1051 DF, p-value: < 2.2e-16
```

## 9. Sem outliers

## 10. Sumário

Veja que na segunda imagem (9) que sem aqueles valores o modelo centra-se nas vendas feitas na linha de 1000 Units Sold que é bem larga e é desbalanceada de uma reta totalmente horizontal pelo grosso de valores na parte inferior. O fato da reta não passar mais pelo centro dos valores inferiores indica um desbalanceamento por falta dos valores superiores que ficavam mais a frente no eixo da abscissa.



## 4 Referências

Sales of summer clothes in E-commerce Wish. **Kaggle**, 2020. Disponível em:  
<<https://www.kaggle.com/jmmvutu/summer-products-and-sales-in-ecommerce-wish>>.  
Acesso em: 3 de set. de 2020.