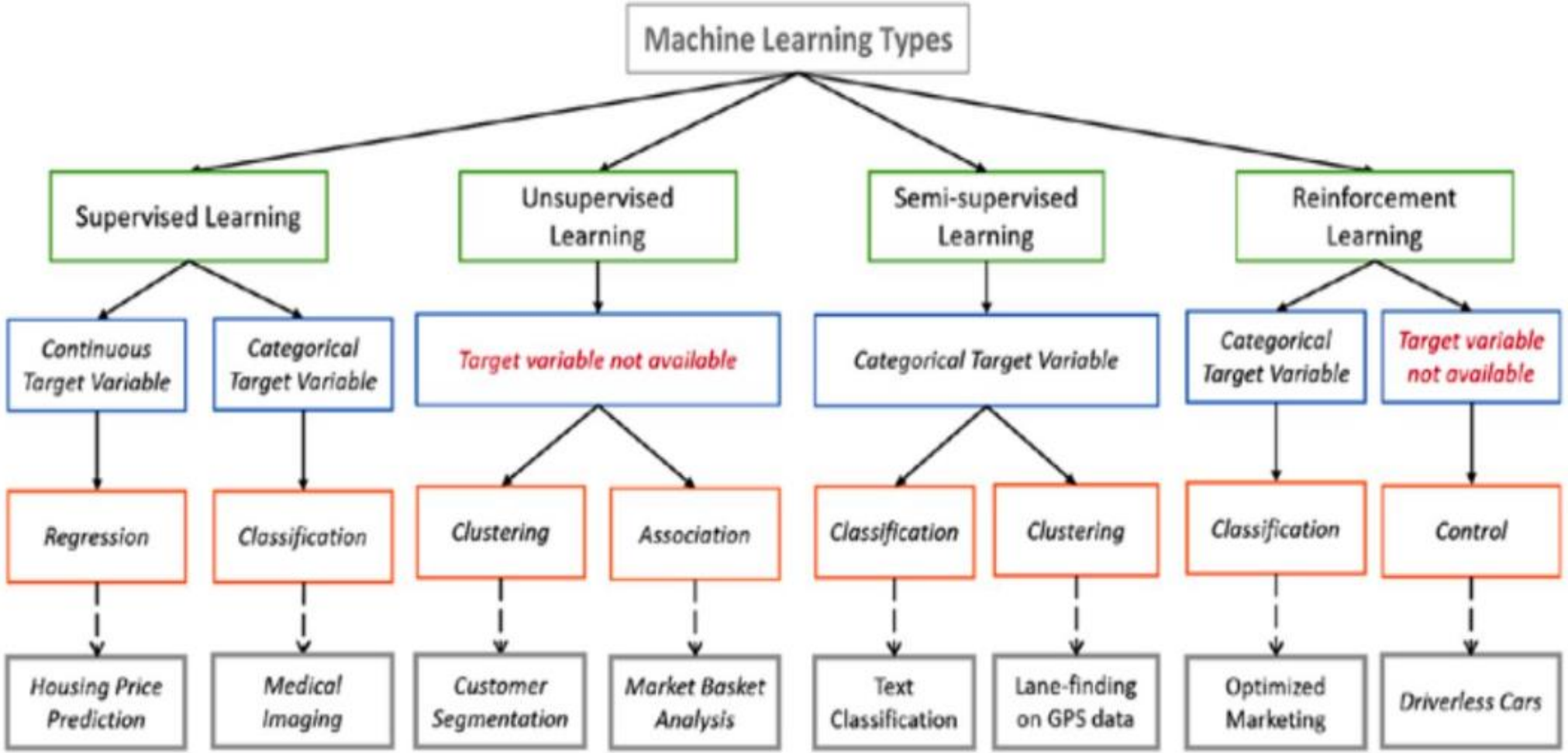
Regressão linear múltipla

Fatec 2025

Tipos de aprendizado de máquina



Variáveis

Variável Independente (ou explicativa, preditora)

- É a variável que **controlamos ou observamos** para explicar ou prever outra
- Representa a **causa** ou o fator que influencia a outra variável
- Fica do lado direito da fórmula do modelo

◆ **Exemplo**: horas_estudo

Quanto mais horas a pessoa estuda, maior tende a ser sua nota.

Variável Dependente (ou resposta, regressanda)

- É a variável que **queremos prever ou explicar**
- Representa o **efeito** ou o resultado influenciado pela variável independente.
- Fica do lado esquerdo da fórmula do modelo.

◆ **Exemplo**: nota_prova

A nota depende das horas de estudo.

Lm()-Linear Model – Modelo linear

- A função lm() ajusta modelos lineares, como **regressão linear simples** ou **múltipla**, usando o método dos **mínimos quadrados**
- Sintaxe:
  - `lm(formula, data, subset, weights, na.action, method = "qr", ...)`

Argumento	Descrição
`formula`	A equação do modelo. Ex: `y ~ x1 + x2` significa "y em função de x1 e x2"
`data`	Data frame onde as variáveis estão definidas
`subset`	Subconjunto dos dados a ser usado
`weights`	Pesos para os valores das observações
`na.action`	Como tratar valores ausentes (NA)
`method`	Método usado para calcular (geralmente "qr" por padrão)

Retorno:

Um objeto da classe "lm" que contém:

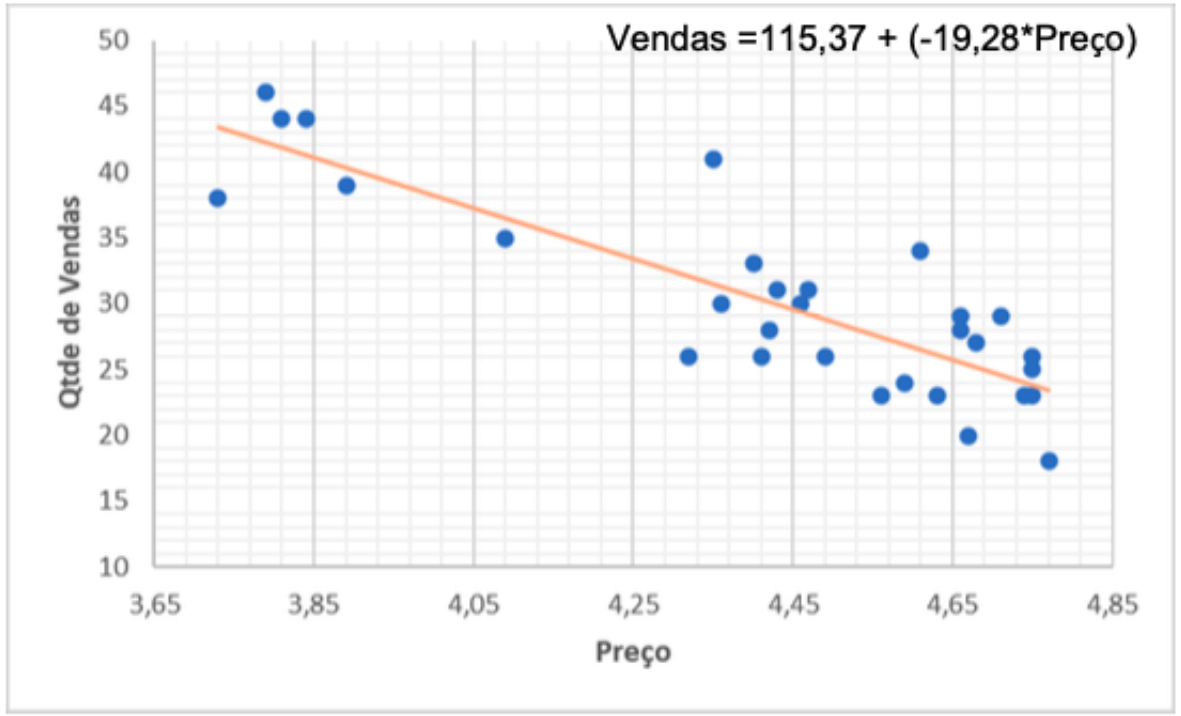
- Coeficientes do modelo (intercepto e inclinações)
- Resíduos
- Valores ajustados (fitted)
- Informações de diagnóstico

Regressão linear

- A Regressão Linear permite gerar um modelo matemático através de uma reta que explique a relação linear entre variáveis
- No caso mais simples, teremos a relação entre uma variável explicativa X (independente) e uma variável resposta Y (dependente)
- Outro fator bastante positivo da regressão linear, e que faz dela um algoritmo
- muito utilizado tanto na estatística “tradicional” quanto em machine learning, é que o algoritmo nos traz informações inferenciais, ou seja, podemos extrapolar conclusões para a população a partir da amostra e também podemos utilizá-lo para modelagem preditiva

Regressão linear Simples

- A regressão linear simples é quando temos apenas uma variável preditora e uma variável resposta



$$Vendas\ do\ Café = 115{,}37 + (-19{,}28 * Preço\ do\ Café)$$

Onde 115,37 é o intercepto B_0 e -19,28 é o coeficiente angular B_1. Ou seja, a cada real que aumenta no preço, as vendas caem, em média, em 19,28 unidades.

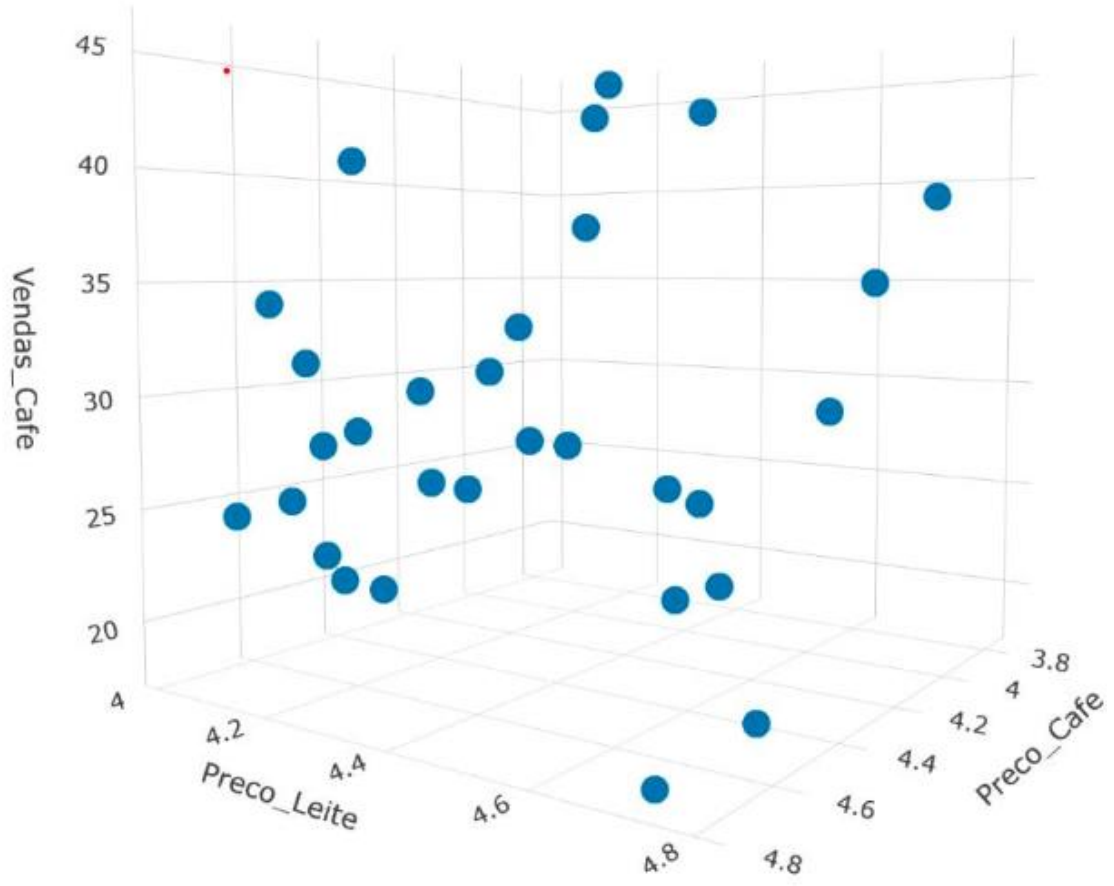
Regressão linear múltipla

- Quando se inclui mais uma variável independente
- Por exemplo, suspeita-se que na nossa empresa, o leite seja um produto complementar ao café, ou seja, o cliente que compra café tende a comprar leite
  - Se essa hipótese for verdadeira, ao realizar mudanças no preço do leite as vendas do café também poderão ser impactadas
  - Ao incluir mais uma variável no modelo (preço do leite) estamos trabalhando com três variáveis:

    duas variáveis independentes (preço do café, preço do leite) e

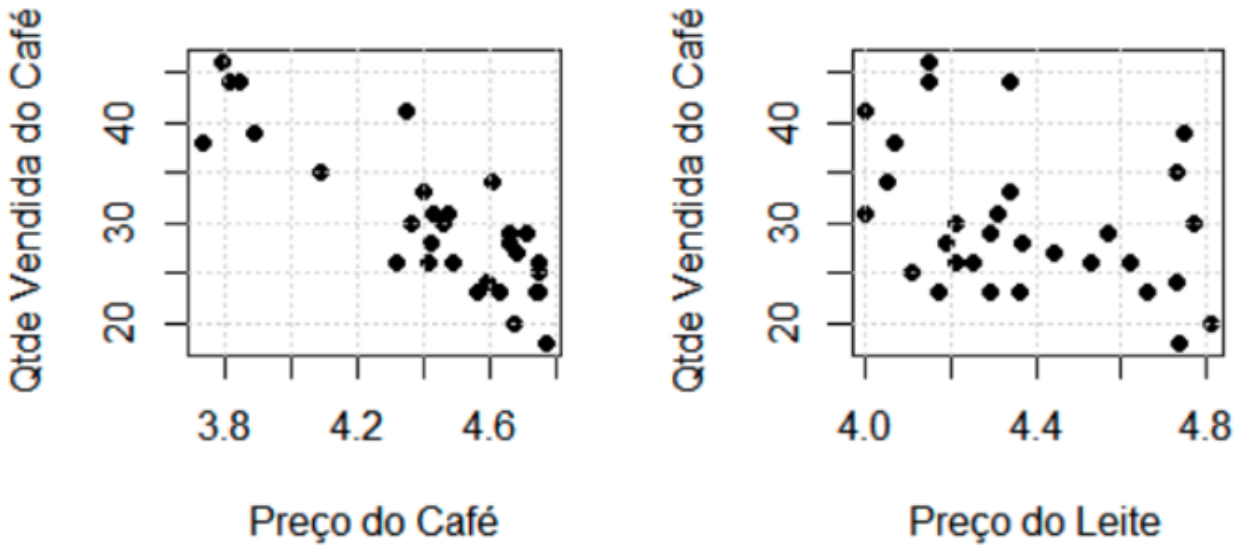
    uma variável dependente (quantidade vendida do café)

Regressão linear múltipla



É possível ver que quando o preço do leite aumenta as vendas do café caem,
assim como quando o preço do café aumenta as vendas do café também caem.

Regressão linear múltipla



A equação de regressão múltipla ficou:

$$Vendas\ do\ Café = 151{,}31 + (-18{,}72 * Preço\ do\ Café) + (-8{,}78 * Preço\ do\ Leite)$$

Projeto Cafeteria

```r
# Instala e carrega os pacotes
install.packages("effects") #, dependencies=TRUE)
#install.packages("ggplot2") #, dependencies=TRUE)

library(effects)
library(ggplot2)
```

```r
# Carregar os dados a partir de um arquivo CSV
dados <- read.csv("dados_cafeteria.csv")
# Ver os primeiros registros
head(dados)
```

Projeto Cafeteria : Estatística descritiva

```r
### Conhecendo os dados ###
# Como estão distribuídos os preços e as vendas?

summary(dados)
sd(dados$Vendas_Cafe)
sd(dados$Preco_Cafe)
sd(dados$Preco_Leite)

# Visualizando a distribuição das variáveis
par(mfrow=c(2,2))
hist(dados$Vendas_Cafe, col='skyblue', main='Distribuição das Vendas de Café')
hist(dados$Preco_Cafe, col='salmon', main='Distribuição dos Preços do Café')
hist(dados$Preco_Leite, col='lightgreen', main='Distribuição dos Preços do Leite')
par(mfrow=c(1,1))
```

Projeto Cafeteria

```
### Preço influencia vendas? ###
# Observar visualmente a relação entre preço e quantidade vendida.

plot(dados$Preco_Cafe, dados$Vendas_Cafe,
     pch = 16,
     col = 'blue',
     xlab = 'Preço do Café',
     ylab = 'Vendas de Café',
     main = 'Preço vs Vendas de Café')
grid()
```

A função grid() no R **desenha linhas de grade no gráfico** quando você está usando **funções do sistema gráfico grid**, como grid.rect(), grid.text(), grid.points() etc.

Projeto Cafeteria

```r
### E as promoções, fazem diferença? ###
# Pontos em vermelho quando há promoção.

cores <- ifelse(dados$Promocao == "Sim", "red", "black")
plot(dados$Preco_Cafe, dados$Vendas_Cafe,
     col = cores,
     pch = 16,
     xlab = 'Preço do Café',
     ylab = 'Vendas de Café',
     main = 'Impacto da Promoção nas Vendas')
legend("topright", legend=c("Com Promoção", "Sem Promoção"),
       col=c("red", "black"), pch=16)
grid()
```

Projeto Cafeteria

```r
### Acima ou abaixo da média? ###
# Quantas vezes as vendas ficaram acima ou abaixo da média histórica?

media <- mean(dados$Vendas_Cafe)
variavel <- ifelse(dados$Vendas_Cafe > media, 'Acima da média', 'Abaixo da média')
variavel <- factor(variavel)
plot(variavel, col=c('gray','orange'), main='Vendas em relação à média')
table(variavel)
```

função table() cria uma **tabela de frequências absolutas** — ou seja, ela **conta quantas vezes cada valor único aparece em uma variável**

Projeto Cafeteria

```
### Diferença entre dias com e sem promoção ###
boxplot(dados$Vendas_Cafe ~ dados$Promocao,
        col = 'gray',
        pch = 16,
        xlab = 'Promoção',
        ylab = 'Vendas',
        main = 'Vendas com Promoção vs Sem Promoção')
```

Projeto Cafeteria: Regressão Linear

📌 **Argumentos mais usados em legend():**

Argumento	Significado
`col`	**Cor** das linhas ou pontos da legenda (ex: `"red"`, `"blue"`, 1, 2, etc.)
`pch`	**Símbolo de ponto** (point character) usado nos gráficos e na legenda (ex: 1 = círculo, 16 = ponto cheio, etc.)
`lty`	**Tipo de linha** (line type), usado para gráficos de linha (ex: 1 = linha contínua, 2 = tracejada, etc.)
`lwd`	**Espessura da linha** (line width), ex: 1, 2, 3

```r
### Regressão linear simples ###
# Vamos modelar a relação entre preço do café e vendas.
modelo <- lm(Vendas_Cafe ~ Preco_Cafe, data = dados)
summary(modelo)

# Gráfico com linha de regressão
plot(dados$Preco_Cafe, dados$Vendas_Cafe,
     col = cores,
     pch = 16,
     xlab = "Preço do Café",
     ylab = "Vendas",
     main = "Linha de Regressão: Preço vs Vendas")
abline(modelo, col = "blue", lwd = 2)
legend("topright",
       legend = c("Promoção", "Sem Promoção", "Linha de Regressão"),
       col = c("red", "black", "blue"),
       pch = c(16, 16, NA),
       lty = c(NA, NA, 1),
       lwd = c(NA, NA, 2))
grid()
```

Projeto Cafeteria: Regressão linear Múltipla

```r
# Convertendo a variável 'Promocao' para fator (variável categórica)
dados$Promocao <- factor(dados$Promocao)

# Ajuste da regressão múltipla com 3 variáveis explicativas:
# Preco_Cafe (quantitativa), Promocao (categórica), Preco_Leite (quantitativa)
modelo_multiplo <- lm(Vendas_Cafe ~ Preco_Cafe + Promocao + Preco_Leite, data = dados)

# Resumo da regressão múltipla
summary(modelo_multiplo)
```

Sem factor(), R trata as categorias como texto comum (strings), o que **impede muitos tipos de análise estatística ou plotagem automática**
Com factor(), o R entende que ”SIM" e ”NÃO" são categorias distintas, não valores contínuos

Uma **variável dummy** (ou **variável indicadora**) é uma variável numérica binária que representa categorias qualitativas com **valores 0 ou 1**.
É usada em modelos estatísticos, como **regressão linear múltipla**, para **incluir variáveis categóricas** (como sexo, cor, região) na análise, já que esses modelos requerem variáveis numéricas

Projeto Cafeteria

```r
# Gera os efeitos marginais estimados do modelo múltiplo
efeitos <- allEffects(modelo_multiplo)
```

```r
# Ajusta layout: 2 linhas por 2 colunas, com espaço para título externo (parte superior)
par(mfrow = c(2, 2), oma = c(0, 0, 3, 0))  # oma = margem externa: top = 3 linhas

# Plota os efeitos com ggplot2 para melhor visualização
plot(efeitos,
     multiline = TRUE,
     ci.style = "bands",  # bandas de confiança
     main = "Variáveis x Vendas Café")
```

A função allEffects() , do pacote **effects**, calcula e organiza **os efeitos principais e de interação de um modelo de regressão** em um formato que facilita a **visualização e interpretação**

Projeto Cafeteria

```
#. Visualizar Interações
Se houver uma interação no modelo (ex: Preco_Cafe * Promocao),
você pode visualizar como o efeito do preço varia com a promoção ativa ou não:

# Corrigir a variável Promocao pois effects não aceita var.categóricas, só factor
dados$Promocao <- factor(dados$Promocao, levels = c("Nao", "Sim"))

# Ajustar modelo novamente
modelo <- lm(Vendas_Cafe ~ Preco_Cafe * Promocao + Preco_Leite, data = dados)

# calcular o efeito da interação
efeito_interacao <- Effect(c("Preco_Cafe", "Promocao"), modelo)

# Plotar
plot(efeito_interacao, multiline = TRUE, ci.style = "bands")

#a inclinação da linha para “Sim” vs “Não” mostra como a efetividade do preço
# muda com a promoção.
```

Projeto Cafeteria

```r
#Analisando somente o efeito de uma variável
modelo <- lm(Vendas_Cafe ~ Preco_Cafe * Promocao + Preco_Leite, data = dados)
efeito_preco <- Effect("Preco_Cafe", modelo)
plot(efeito_preco, multiline = TRUE, ci.style = "bands")

#“Para diferentes valores de Preco_Cafe, qual é a estimativa de Vendas_Cafe,
# considerando que Promocao está em um nível fixo (ex: ‘Nao’) e
#Preco_Leite está em seu valor médio?”
```

Projeto Cafeteria

```r
modelo_completo <- lm(Vendas_Cafe ~ Preco_Cafe + Preco_Leite + Promocao, data = dados)
modelo_sem_promocao <- lm(Vendas_Cafe ~ Preco_Cafe + Preco_Leite, data = dados)

# Comparar efeitos marginais
plot(allEffects(modelo_completo), main = "Com Promoção")
plot(allEffects(modelo_sem_promocao), main = "Sem Promoção")


#Mesmo sob a condição de promoção ativa:
# O preço do café ainda impacta fortemente as vendas (quanto menor, mais vende);
# O preço do leite também afeta negativamente as vendas de café,
# possivelmente por serem consumidos juntos;
#A própria promoção tem um efeito positivo nas vendas, como esperado.
#Os gráficos representam efeitos marginais condicionais e mostram como cada variável
#influencia a resposta mantendo as outras fixas (Promocao = "Sim").
```

Projeto Floricultura

- Você foi contratado por uma floricultura especializada em espécies do gênero *Iris*. A empresa deseja entender como diferentes características das flores se relacionam entre si, a fim de prever o **comprimento das pétalas** a partir de outras medições físicas.
- Para isso, você terá acesso a uma base de dados clássica da estatística chamada **iris**, que contém medições de 150 flores de três espécies (*setosa*, *versicolor* e *virginica*). Cada flor foi medida em relação a:
  - **Sepal.Length** – comprimento da sépala (em cm)
  - **Sepal.Width** – largura da sépala (em cm)
  - **Petal.Length** – comprimento da pétala (em cm)
  - **Petal.Width** – largura da pétala (em cm)
  - **Species** – espécie da flor

Projeto floricultura

- Qual variável teve o maior impacto sobre o comprimento da pétala: o comprimento da sépala ou a largura da sépala?
- O modelo parece se ajustar bem visualmente aos dados?
- Como você interpretaria esse modelo para um colega da floricultura que não conhece estatística?

Projeto floricultura

```r
install.packages("effects")
install.packages("scatterplot3d")
```

```r
# Carregar a base iris
data(iris)

# Visualização geral
head(iris)
str(iris)
summary(iris)

# Tabela de frequência para a variável categórica
table(iris$Species)

# Gráficos exploratórios
boxplot(Petal.Length ~ Species, data = iris, main = "Petal Length por Espécie", col = "lightblue")
pairs(iris[1:4], main = "Matriz de dispersão")
```

pairs()

- Mostra **visualmente a correlação** entre pares de variáveis.
- Ajuda a **identificar relações lineares, padrões e outliers**.
- Útil como **análise exploratória inicial** em conjuntos de dados multivariados.

Projeto floricultura

- A biblioteca **scatterplot3d** é usada para criar **gráficos de dispersão em 3 dimensões** (3D)
- útil para **visualizar relações entre três variáveis numéricas** em modelos exploratórios e até mesmo para ilustrar planos de regressão linear múltipla.

```
library(scatterplot3d)

# Atribuir cores por espécie
colors <- as.numeric(iris$Species)

scatterplot3d(iris$Sepal.Length,
              iris$Sepal.Width,
              iris$Petal.Length,
              color = colors,
              pch = 16,
              xlab = "Sepal Length",
              ylab = "Sepal Width",
              zlab = "Petal Length",
              main = "Gráfico 3D Estático com scatterplot3d")
legend("topright", legend = levels(iris$Species), col = 1:3, pch = 16)
```

Projeto Floricultura

```r
# ------------------------
# Regressão Linear Simples
# ------------------------

modelo_simples <- lm(Petal.Length ~ Sepal.Length, data = iris)
summary(modelo_simples)

# Gráfico com a reta de regressão
plot(iris$Sepal.Length, iris$Petal.Length,
     main = "Petal.Length vs Sepal.Length",
     xlab = "Sepal Length", ylab = "Petal Length", pch = 16)
abline(modelo_simples, col = "blue", lwd = 2)
grid()
```

Projeto Floricultura

```r
# ----------------------
# Regressão Linear multipla
# ----------------------
# Ajustar modelo de regressão linear múltipla
modelo_multiplo <- lm(Petal.Length ~ Sepal.Length + Sepal.Width, data = iris)
summary(modelo_multiplo)
# Extrair coeficientes
coef <- round(coef(modelo_multiplo), 2)
eq_text <- paste0("Petal.Length = ", coef[1], " + ",
                  coef[2], " * Sepal.Length + ",
                  coef[3], " * Sepal.Width")
# Criar gráfico 3D com pontos
grafico <- scatterplot3d(iris$Sepal.Length, iris$Sepal.Width, iris$Petal.Length,
                         pch = 16, color = as.numeric(iris$Species),
                         xlab = "Sepal Length", ylab = "Sepal Width", zlab = "Petal Length",
                         main = "Regressão Linear Múltipla: Petal.Length ~ Sepal.Length + Sepal.Width")
# Adicionar plano de regressão
grafico$plane3d(modelo_multiplo)
# Adicionar a equação no gráfico (posição X, Y, Z ajustável)
text(x = 4.5, y = 4.2, labels = eq_text, cex = 0.8, pos = 4)
```

Projeto Floricultura

```r
# Comparação com o modelo anterior
anova(modelo_simples, modelo_multiplo)
```

Coluna	Significado
`Res.Df`	Graus de liberdade residual de cada modelo
`RSS`	Soma dos quadrados dos resíduos (erro)
`Df`	Diferença nos graus de liberdade entre os modelos
`Sum of Sq`	Redução na soma dos quadrados (ganho explicativo)
`F`	Estatística F calculada para comparar os modelos
`Pr(>F)`	Valor-p: se for **pequeno (< 0.05)**, o modelo mais complexo é significativamente melhor

Neste exemplo, observe o RSS (erro)

Projeto Floricultura

```r
# ------------------------
# Inclusão de variável categórica e efeitos
# ------------------------

is.factor(iris$Species)  # Deve retornar TRUE

modelo_species <- lm(Petal.Length ~ Sepal.Length + Species, data = iris)
summary(modelo_species)

# Visualização dos efeitos
library(effects)
efeitos <- allEffects(modelo_species)
plot(efeitos)
```

Atividade: concessionária

- Você trabalha em uma **concessionária de carros** e deseja entender como características técnicas dos veículos influenciam o **consumo de combustível (mpg - miles per gallon)**
- Você tem à disposição a base de dados mtcars, que contém informações de 32 modelos de carro, com variáveis como:
  - mpg: consumo (milhas por galão)
  - hp: potência do motor (horsepower)
  - wt: peso do carro
  - cyl: número de cilindros
  - am: tipo de transmissão (0 = automática, 1 = manual)

Atividade

- **Parte 0 – Estatística descritiva**
- Use summary(mtcars) e str(mtcars) para explorar a base.
- Gere gráficos de dispersão com plot() ou pairs().
- **Parte 1 – Regressão Linear Simples**
- Ajuste um modelo de regressão linear simples prevendo mpg a partir de hp
- **Parte 2 – Regressão Linear Múltipla**
- Adicione o peso do carro como segunda variável

Atividade

- **Parte 3 – gráfico 3D**
  - Existe algum ponto extremo (outlier) que parece não se ajustar bem à reta/plano de regressão?

- **Parte 4 – Interpretação**
- Responda:
  - O que significa o sinal dos coeficientes?
  - O modelo com duas variáveis é melhor que o modelo simples? Justifique usando o anova()
  - Que variáveis poderiam ser incluídas para melhorar o modelo?