Uma Introdução à Ciência de Dados com uso de R

Murielly Oliveira Nascimento

Universidade Federal de Uberlândia
Faculdade de Computação
Projeto de Iniciação Científica do PET-SI.

Uberlândia
2021

Murielly Oliveira Nascimento

Uma Introdução à Ciência de Dados com uso de R

Dissertação apresentada ao Programa de Educação Tutorial(PET) da Faculdade de Sistemas de Informação da Universidade Federal de Uberlândia como parte dos requisitos para a participação do programa PET.

Área de concentração: Sistemas de Informação

Orientador: Flávio de Oliveira Silva
Coorientador: Wendel Alexandre Xavier de Melo

Uberlândia
2021

Este trabalho é dedicado à todas as mulheres que,
um dia sonharam em alcançar o impossível.

Agradecimentos

Agradeço minha família pelo apoio e meu orientador,Flávio Silva de Oliveira, pelos inestimáveis conselhos e conhecimentos passados. Ao tutor do PET-SI, Wendel Melo,meus cumprimentos por inspirar todos os membros do programa em tempos de pandemia. Por fim,meu apreço pela universidade UFU por tornar esse trabalho possível.

"Quanto mais estudo, mais sinto que minha mente nisso é insaciável."
(Ada Lovelace)

Resumo

O presente trabalho tem como objetivo introduzir o leitor ao tema Ciência de Dados, do que se trata, qual o cenário atual e as expectativas para o futuro. Também são apresentados exemplos práticos em R a fim de consolidar o conhecimento obtido.

Com esse intuito foram usadas diversas fontes durante a pesquisa, dentre elas as principais foram, Introdução à Ciência de Dados por Fernando Amaral e os materiais disponibilizados pela empresa IBM no site Coursera.A exemplo da primeira este texto começa fornecendo alguns conceitos da área ao leitor para em seguida mergulhar no ciclo de vida do objeto de estudo da Ciência de Dados, o próprio dado.

Uma vez compreendidos os procedimentos para produção, armazenamento, transformação, análise e descarte parte-se para a parte prática com alguns exemplos de algoritmos e códigos em R. São usadas plataformas como JupyterLab e ferramentas como Weka Open Source R.

Conclui-se reforçando ao leitor os benefícios deste texto à comunidade científica e oferecendo recomendações para aqueles que desejarem se aprofundar no tema.

Palavras-chave: Dados. Big Data. Ciência de Dados. Mineração de Dados. .

Abstract

This paper aims to introduce the reader into the field of Data Science, what it is about, the current scenario and the expectations for the future. Practical examples in R are also presented in order to consolidate the knowledge obtained.

To this end, several sources were used during the research, amongst them the main ones were, Introduction to Data Science by Fernando Amaral and the materials made available by the IBM company on the Coursera website. In the same way Fernando Amaral does, this paper starts with the life cycle of data.

Once the procedures for production, storage, transformation, analysis and disposal of data are understood, the practical part starts with some examples of algorithms and codes in R. Platforms like JupyterLab and tools like Weka Open Source R are used.

This paper concludes by reinforcing to the reader the benefits of the text to the scientific community and recommendations for those who wish to delve deeper into the theme.

Keywords: Data. Big Data. Data Science. Data Mining.

Lista de ilustrações

Lista de siglas

DepDelayMinutes Atraso em minutos da partida

EQM Erro Quadrático Médio

LGPD Lei Geral de Proteção aos Dados

LateAircraftDelay Atraso causado por outro avião

Sumário

Capítulo 1

Introdução

O dado como objeto de estudo nada mais é do que fatos coletados que quando interpretados geram informação, e a partir dela conhecimento. Ciência de Dados se propõe a estudar o dado durante todo o seu ciclo de vida, começando pelo armazenamento e indo até o seu descarte.

Embora esse ramo de estudo exista desde a década de 60,apenas recentemente entrou em voga. Isso se deve, segundo Fernando Amaral, ao barateamento,miniaturização e aumento da capacidade de processamento que levaram à disseminação de equipamentos capazes de armazenar e produzir dados.Tais mudanças são englobadas no fenômeno conhecido como Big Data.

Para alguns ele é considerado como a 4a fase da Revolução Industrial, transformando como empresas negociam e se relacionam com seus clientes e como a sociedade se estrutura de forma geral. Diferente das metodologias tradicionais onde empresas usam um data warehouse e ficam limitadas a dados estruturados, o escopo de Big Data é bem mais amplo, permitindo a análise de dados indiretamente relacionados com um negócio.

> Big Data está trazendo mudanças profundas na indústria. Na produção, Big Data vai ser capaz de tornar os processos produtivos mais eficientes: menores custos, maior produção, períodos de paradas não programadas menores. Na área administrativa, Big Data vai permitir que haja menos fraude, menos desperdício, menos passivos judiciais, menos pagamento de impostos. No relacionamento com os clientes, melhor fidelização, mais qualidade, clientes mais satisfeitos. Big Data também vai mudar a relação das empresas com seus fornecedores e parceiros comerciais. Big Data é a nova revolução industrial, sua 4ª fase.(AMARAL, 2016)

Um hotel,por exemplo, pode usar informações sobre seus clientes, estas publicadas em suas redes sociais, para prever o movimento que terá em determinado mês do ano. Internamente, a empresa pode realizar processos seletivos mais eficientes, elegendo candidatos com o perfil adequado a uma vaga e a cultura da empresa.

Portanto, o fenômeno Big Data e o estudo de Ciência de Dados, são temas essenciais que prometem transformar a sociedade como um todo. O conhecimento destes assuntos é benéfico a qualquer parte interessada.

1.1 Motivação

Nos anos de 2019 e 2020 foram lançados os documentários Privacidade Hackeada e O Dilema das Redes,respectivamente. Em ambos o dado é abordado como sendo um dos commodities mais rentáveis do século XXI. No segundo, Shoshana Zuboff, professora em Havard, afirma que empresas especializadas no tratamento de dados vêm produzindo os trilhões de dólares que fizeram companhias de tecnologia as mais ricas na história da humanidade.

A empresa Cambridge Analytica é considerada uma das grandes responsáveis pela eleição do ex presidente dos Estados Unidos Donald Trump. E o seu trabalho na campanha dele foi reunir informações de eleitores americanos, coletados pelo Facebbok, que estavam indecisos sobre seu voto e,portanto, suscetíveis a influências externas.

Nessas situações observa-se a relevância de compreender como o dado é produzido e usado, principalmente por grandes empresas como Google e Facebook. Para o público geral é relevante entender como os seus dados são usados e quais os seus direitos sobre eles.

No caso de profissionais que trabalham direta ou indiretamente com o tratamento de dados, há a demanda mercadológica pelo estudo do assunto. O conhecimento aplicado de Ciências de Dados, pode, em um hospital ajudar a reconhecer câncer em seus estágios iniciais, em empresas a atender melhor seus clientes,em academias calcular o progresso de um atleta,etc.

1.2 Objetivos e Desafios da Pesquisa

A pesquisa tem como objetivo uma introdução ao estudo da Ciência de Dados, metodologia, ferramentas utilizadas e perspectivas futuras. O trabalho se justifica devido a importância para a sociedade do tema e a demanda crescente por profissionais capacitados nele.

Em uma entrevista com a empresa IBM Shingai Manjengwa, gerente executiva da Fireside Analytics, afirma que o cientista de dados, os usa para entender o mundo. Outra reportagem define o profissional como um dos mais desejados do século 21.

> Se “sexy” significa ter qualidades raras que são muito procuradas, os cientistas de dados já estão lá. Eles são difíceis e caros de contratar e, devido ao mercado muito competitivo para seus serviços, difíceis de reter. Simplesmente não há muitas pessoas com sua combinação de formação científica e habilidades computacionais e analíticas.(DAVENPORT; PATIL, 2012).

Trata-se, portanto, de uma área de estudo em crescente demanda, cujos muitos setores em que é aplicada estão passando por dificuldades devido à falta de mão de obra

qualificada. Considerando a tendência dos dados aumentarem exponencialmente a cada segundo, é essencial que profissionais de TI sejam instruídos no tópico.

Além disso, recentemente foi aprovada a Lei Geral de Proteção aos Dados (LGPD) e considerando o valor atribuído à informação atualmente, as empresas mais valorizadas são aquelas que armazenam e produzem dados(Google, YouTube,Facebook...), passa ser fundamental que o público geral entenda o tema. Como se dá a produção, armazenamento e processamento de dados, este trabalho se propõe a responder essas perguntas.

> Art. 1º Esta Lei dispõe dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural.(LGPD. . . , 2018).

Logo, Ciência de Dados é um ramo de estudo novo, porém a caminho de se tornar um dos pilares da sociedade. A sua aplicação é essencial a empresas e a formação de profissionais nela também. O público dentro e fora do meio acadêmico também se beneficia deste conhecimento, como demonstrado anteriormente.

1.3 Organização da Dissertação ou Tese

No capítulo 2 é introduzido ao leitor os conceitos fundamentais para a compreensão do tema. No caso, o ciclo de vida do dado, cujas etapas são: produção, armazenamento, transformação, análise e descarte. A análise de dados é desdobrada em análise implícita e explícita.

No capítulo 3 são apresentados exemplos práticos em linguagem R do tema,os dados usados foram extraídos do livro Introdução à Ciência de Dados. Algumas abordagens feitas também foram baseadas no material da empresa IBM. Além disso,plataformas como JupyterLab e as ferramentas Weka e Open Source R são utilizadas.

Conclui-se,no capítulo 4, reforçando os benefícios desta pesquisa à comunidade científica. Apresentado recomendações para o leitor aprofundar-se no tema e perspectivas futuras para esta área de estudo.

Capítulo 2

Fundamentação Teórica

2.1 Ciclo de Vida do Dado

Figura 1 – Ciclo de vida do dado

Fonte: Autoria Própria

A primeira etapa no ciclo de vida do dado é a sua produção. O mesmo pode ser obtido desde as formais mais convencionais (a entrada de dados pelo teclado de um computador) até os meios usados pelo projeto SETI (sinais de rádio são captados do espaço para averiguar a possibilidade de vida extraterrestre). De forma geral, a produção de dados se dá através de sensores,pelo processamento e análise dos mesmos, ou pela transformação.

Quanto ao seu armazenamento houve uma série de mudanças desde o primeiro modelo na década de 60. Nesta época, apelidada de pré-relacional o modelo hierárquico e em rede se popularizaram. A ideia então era armazenar dados em registros vinculados,pelos quais se navegava a fim de recuperar uma informação.

Por volta de 1970 Edgar Frank Codd criou o modelo relacional, o mais eficiente para aplicações de negócio e utilizado até hoje em situações nas quais se busca manter a integridade do dado. Em 1980 o modelo Orientado a Objetos surgiu, cujo objetivo era suportar as linguagens de programação desse paradigma, como C++ e Delphi.

Por fim em 2000 o modelo NoSQL passou a ser implementado em situações onde os dados não eram normalizados e o foco era processamento, escalabilidade e volume. Contudo, este não substitui os seus predecessores, apenas é mais adequado a um tipo de situação. Sendo esta, a de tratar dados não estruturados.

Dados semiestruturados costumam vir em planilhas, facilitando a sua normalização. Dados não estruturados, por sua vez, podem estar em diversos formatos como áudio, imagens, posts em blogs,e-mails,etc. Segundo Fernando Amaral de 80% a 90% dos dados produzidos ao redor do mundo não são estruturados.

Os modelos anteriormente mencionados, são apropriados para o armazenamento de dados não para a sua análise. Por isso em 1990 criou-se grandes depósitos de dados chamados data warehouses, estruturados a partir de modelos relacionais. Sua diferença estava na forma de armazenamento, em outras palavras, os dados já viam pré-calculados e não normalizados. Diferente de sistemas transacionais eles guardam um histórico dos dados.

São,entretanto, difíceis e caros de construir. A quantidade enorme de dados produzidos hoje agrava o problema. O modelo MapReduce, criado por funcionários do Google, é uma forma simples de processar grandes volumes de dados,dividindo o processamento entre computadores em redes. Cada computador é, então, um nó na rede. Hadoop é a implementação mais conhecida desse modelo.

A análise de dados pode ser dividida em três categorias diferentes: exploratória, explícita e implícita. A primeira tem como objetivo conhecer os dados, antes de aplicar qualquer técnica sobre eles.Análises explícitas, por sua vez, destacam informações preexistentes em conjuntos de dados. Por fim, as análises implícitas buscam informações que não estão claras em um conjunto de dados e que para obtê-las deve usar-se uma técnica mais sofisticada.

O descarte de dados é um tema de constante debate desde a criação e implementação de leis como a LGPD. Hoje há empresas que faturam milhões com a coleta e tratamento de dados. Em contrapartida, clientes dessas mesmas companhias demandam a privacidade e controle de suas informações. Um post feito no site da Universidade de Missouri afirma que a única forma de apagar dados completamente de um computador seria destruindo-o.

Capítulo 3

Análises Práticas com R

3.1 Introdução à Linguagem R

A linguagem R é *sensitive case*, ou seja, diferencia letras maiúsculas de minúsculas, sendo comum o uso da sintaxe *camelcase*. Nela, as funções começam com a primeira letra minúscula e as demais maiúsculas.

Considerando o ambiente de programação JupyterLab a maioria dos pacotes necessários já estão instalados. Localmente é preciso baixá-los primeiro. Para isso usa-se a função *install.packages(Parâmetro 1, Parâmetro 2)*. O primeiro parâmetro recebe o nome do pacote, o segundo é do tipo booleano que define se as dependências de um pacote serão instaladas ou não. Outra observação importante é que todos os códigos aqui apresentados podem ser encontrados no repositório GitHub da autora.

```
# Caso queira instalar o pacote ggplot2
# install.packages("ggplot2",dependencies=TRUE)

# Essa funcao verifica os pacotes instalados na IDE
installed.packages()

# Carregando pacotes
library(datasets)

# Verificando os pacotes carregados
search()

# E bom descarregar um pacote quando ele nao for mais necessario,
# pois uma quantidade massiva deles pode fazer diferenca.
# Principalmente por serem mantidos na memoria.
detach("package:datasets")
search()
```

R oferece diversas funções previamente construídas em suas bibliotecas. Não há necessidade de decorar os parâmetros e funcionamento de cada uma delas. Em caso de dúvida,

usa-se ? e o nome da função para obter mais informações sobre ela.

3.1.1 Tipos de Dados

Para atribuir valores a uma variável pode-se uar o operador <- ou =. Os tipos de dados mais comuns são *caracter,numeric e integer*. É possível verificar a classe de uma variável usando a função *class()* ou *is.numeric()*. Caso queira garantir que ela receba um valor como inteiro use *as.integer()*.

```
var1 <- 10
class(var1)

var1 <- as.integer(10)
class(var1)

var2 <- "Lisboa eh a capital de Portugal"

# Listar os objetos disponiveis no ambiente
objects()

# Para apagar todos os objetos, use a funcao list
# para criar um objeto do tipo lista e apague-o usando rm()
rm(list=objects())
objects()

# Lista de dados pre-carregados do R ou IDE
data()

# Caso queira visualizar uma das datasets basta digitar o seu nome
Titanic
```

3.1.2 Estruturas de Objetos

A linguagem R possui diversas classes que podem armazenar dados em diversas formas e estruturas. Algumas delas são:

Objetos	
Classe	Descrição
Vetores	Conjunto simples de valores do mesmo tipo
Matrizes	Conjunto bidimensional de valores do mesmo tipo
Arrays	Podem ser vetores ou matrizes
Listas	Listas de diferentes objetos, os quais podems ser de tipos diferentes
Data Frames	Parecido com uma tabela de banco de dados
Séries Temporais	Armazena séries de dados temporais
Fatores	Armazena variáveis categóricas

3.2 Análises Iniciais

Antes de analisar um conjunto de dados(dataset) é preciso, primeiramente, carregar a biblioteca *Tidyverse*. O ambiente de programação JupyterLab já vem pré-configurado com ela, de modo que não é preciso baixá-la. Localmente é necessário fazer o download. Algumas funções básicas usadas na análise de um dataset:

mean() Média.

median() Mediana.

var() Variância.

sd() Desvio Padrão

```
# Se estiver executando localmente apague o '#'
# install.packages("tidyverse")

# Carrega tidyverse
library(tidyverse)
```

Os pacotes que vêm nessa biblioteca e auxiliam na análise de dados são:

tidyr ajuda a criar dados organizados.

dplyr manipulação e transformação de dados.

readr faz a leitura de arquivos.

purrr programação funcional e suas ferramentas.

ggplot2 vizualização de dados e gráficos.

3.2.1 Problema

Toda análise de dados busca a solução de um problema, nesse caso, supondo que um funcionário da Google, por exemplo, precise fazer uma viagem de Los Angeles a Nova York com o mínimo de atraso possível. Para isso é necessário saber quais fatores levam ao atraso de um voo, fazer uma estimativa deles e descobrir qual linha aérea é a melhor.

A empresa IBM, mantém em seu site uma série de conjunto de dados. Nesse caso usou-se a amostra do *Airline Dataset* que contém dados de mais de 200 milhões de voos domésticos feitos em território americano, coletados pelo Departamento de Estatísticas de Transporte dos Estados Unidos.

```
# url onde os dados estao localizados
# url <- "https://dax-cdn.cdn.appdomain.cloud/dax-airline/1.0.1/lax_to_jfk.tar.gz"

# download do arquivo
#download.file(url, destfile = "lax_to_jfk.tar.gz")

# extrair arquivo
#untar("lax_to_jfk.tar.gz", tar = "internal")

# Lemos o arquivo csv => read_csv (Se for de um formato diferente use a funcao apropriada. Verifique a documentacao de readr)
# companhias_aereas <- read_csv("lax_to_jfk/lax_to_jfk.csv",
#                       col_types = cols(
#                        'DivDistance' = col_number(),
#                        'DivArrDelay' = col_number()
#                        ))

# Nesse caso, o arquivo ja esta instalado, portanto leremos direto do diretorio dados
companhias_aereas <- read_csv("dados/losAngeles_para_novaYork.csv",
                      col_types = cols(
                       'DivDistance' = col_number(),
                       'DivArrDelay' = col_number()
                       ))
```

Uma vez lido o conjunto de dados, fazer algumas análises iniciais sobre ele.

head(dataframe, n) Retorna as primeiras 6 linhas do conjunto de dados se ‘n‘ não for especificado

tail(dataframe, n) Retorna as últimas 6 linhas do conjunto de dados se ‘n‘ não for especificado

colnames(dataframe) Retorna o nome das colunas

dim(dataframe) Retorna as dimensões do conjunto de dados.

Como mencionado anteriormente, as análises serão sobre os voos feitos de Los Angeles para Nova York, representados respectivamente por LAX e JFK no dataset (conjunto de dados). A sigla JFK faz referência ao Aeroporto Internacional John F. Kennedy. Focando em informações referentes a possíveis atrasos, as colunas ArrDelay(Atraso na chegada), SecurityDelay (Atraso devido a medidas de segurança), CarrierDelay(atrasos que são causados pela companhia aérea como limpeza do avião), e outras são objetos de análises.

```
# A amostra que estamos usando ja vem com os voos LAX para JFK separados
# e com as colunas que precisamos para analise
# As dimensoes e nomes das colunas do dataset sao:
dim(companhias_aereas)
colnames(companhias_aereas)

# Salvar o dataset no diretorio dados
# write_csv(companhias_aereas, "dados/losAngeles_para_novaYork.csv")

# Descobrir o tipo de dados no dataset
sapply(companhias_aereas, typeof)
```

O pacote Dplyr é usado para transformação e manipulação de dados. Algumas de suas funções são:

select() Seleciona.

filter() Filtra.

summarize() Resume.

arrange() Organiza linhas de dados por valores das colunas.

mutate() Adiciona novas variáveis.

group_by() Agrupa dados em um novo dataset.

```
# A media de atrasos na chegada de cada companhia aera
# Os valores na coluna ArrDelay funcionam da seguinte maneira:
# Se o valor eh positivo houve atraso
# Se o valor eh negativo ela chegou adiantada ou cedo.
# O operador '>' pode ser lido como entao

companhias_aereas %>%
  group_by(Reporting_Airline) %>%
  summarize(atraso_Companhia = mean(ArrDelay, na.rm = TRUE))

```

```
# Podemos observar que a empresa com menor tempo medio de atraso na chegada
    eh a AS

# Um resumo do data_set
glimpse(companhias_aereas)
```

3.3 Processamento de Dados

É preciso, antes de analisar um dataset, fazer uma *limpeza dos dados*

3.3.1 Valores NA e Formatação

NA Quando um valor está faltando no dataset a linguagem R atribui o símbolo NA (Not Available) a ele.

NAN Valores impossíveis, como divisão por 0, são representados por NaN(not a number).

Pode-se usar a função 'is.na(x)', para saber se um elemento x é NA.

```
# Contamos a quantidade de valores faltando no dataset
companhias_aereas %>%
  summarize(count = sum(is.na(CarrierDelay)))
```

Usa-se a função 'purrr::map() para aplicar uma função ou fórmula num elemento. O símbolo ' ' separa o lado esquerdo do direito de uma fórmula. Geralmente elas são representadas assim: 'y x', no exemplo a seguir é calculado a soma dos valores NA em cada coluna do dataset.

```
# Aplique a funcao sum(is.na(.)) para cada coluna.
# Em outras palavras, encontre a soma dos valores NA de cada coluna.

map(companhias_aereas, ~sum(is.na(.)))
```

A partir disso, obtém-se as seguintes informações

CarrierDelay 2486 dados faltando

WeatherDelay 2486 dados faltando

NASDelay 2486 dados faltando

SecurityDelay 2486 dados faltando

LateAircraftDelay 2486 dados faltando

DivDistance 2855 dados faltando

DivArrDelay 2855 dados faltando

Além de levar em consideração valores NA, deve-se também verificar se cada dado está no devido formato. No caso deste dataset, deve-se desmembrar a coluna FlightDate nas colunas dia, mês e ano; e modificar os seus tipos para inteiros no lugar de caracteres.

```
# Para futuras analises cada dado deve estar no devido formato
# Aqui pegamos a coluna FlightDate e a desmembramos nas colunas ano, mes e dia
datas_de_voo <- companhias_aereas %>%
    separate(FlightDate, sep = "-", into = c("year", "month", "day"))
```

```
# Por ultimo, observamos que as colunas ano, mes e dia est o representadas como caracteres
# Iremos modifica-los para inteiros
# A funcao abaixo pode ser traduzida como:
# "Modifique todos os valores nas colunas ano,mes e dia se eles forem caracteres"

datas_de_voo %>%
    select(year, month, day) %>%
    mutate_all(type.convert) %>%
    mutate_if(is.character, as.numeric)
```

3.3.2 Normalização de Dados

É o processo de trazer dados distintos para um mesmo *alcance*. Suponha, por exemplo, a comparação de idades de pessoas com seus respectivos salários. Este tipo de dado (salário) invariavelmente tem mais peso num modelo de análise. A fim de solucionar esse problema é preciso usar a distribuição normal. Ela é dada pela seguinte fórmula:

$$x_{new} = \frac{x_{old} - \mu}{\sigma}$$

μ: É a média dos valores

σ: É o desvio padrão

X: É uma variável qualquer.

Geralmente o resultado é uma série de valores variando de 0 a 1.

```
# Usaremos a coluna ArrDelay
# No codigo abaixo mean() calcula a media de ArrDelay e sd() o seu desvio padrao.
# Os dados sao armazenados em escala_Z
# Por fim mostramos os 6 primeiros dados normalizados de ArrDelay na tela

```

```
escala_Z <- (companhias_aereas$ArrDelay - mean(companhias_aereas$ArrDelay))
    / sd(companhias_aereas$ArrDelay)
head(escala_Z)
```

3.3.3 Variável Indicadora

Usadas para aplicação de técnicas de análise em variáveis categóricas. Neste dataset, por exemplo têm-se siglas representando as companhias áreas, é possível substituí-las por números (lembrando que eles serão somente representativos) atribuindo-lhes valores *dummy*.

```
companhias_aereas %>%
  mutate(dummy = 1) %>% # Coluna com um valor
  spread(
    key = Reporting_Airline, # Usamos a coluna Companhias Aereas e a
        espalhamos
    value = dummy,
    fill = 0) %>%
  slice(1:5)
head(escala_Z)
```

```
companhias_aereas %>% # Comecamos com nosso dataset
   mutate(Reporting_Airline = factor(Reporting_Airline,
                                     labels = c("AA", "AS", "DL", "UA", "B6
                                         ", "PA (1)", "HP", "TW", "VX")))%>%
  ggplot(aes(Reporting_Airline)) + #aes() ajuda a criar graficos
      estilizados
  stat_count(width = 0.5) + #ESpessura
  labs(x = "Number of data points in each airline") # Titulo do grafico
```

3.4 Análises Exploratórias

Usadas para entender melhor o dataset. Nesse caso, ajudarão a descobrir as principais causas que levam ao atraso de voos. Para fins de visualização usa-se a biblioteca *ggplot*, responsável por criar gráficos. Suas principais funções são:

3.4.1 Diagrama de Caixa

Também chamado de box plot, o diagrama de caixa é uma ferramenta gráfica em que cada caixa representa uma variável a ser analisada. A mesma destaca os quartis, mediana, maior e menor valor. No código abaixo vê-se a distribuição de atrasos (na chegada) para cada companhia aérea.

```
# Boxplot
ggplot(data = companhias_aereas, mapping = aes(x = Reporting_Airline, y =
    ArrDelay)) +
  geom_boxplot(fill = "bisque",color = "black", alpha = 0.3) +
  geom_jitter(aes(color = 'blue'), alpha=0.2) +
  labs(x = "Airline") +
  ggtitle("Arrival Delays by Airline") +
  guides(color = FALSE) +
  theme_minimal() +
  coord_cartesian(ylim = quantile(companhias_aereas$ArrDelay, c(0, 0.99)))
```

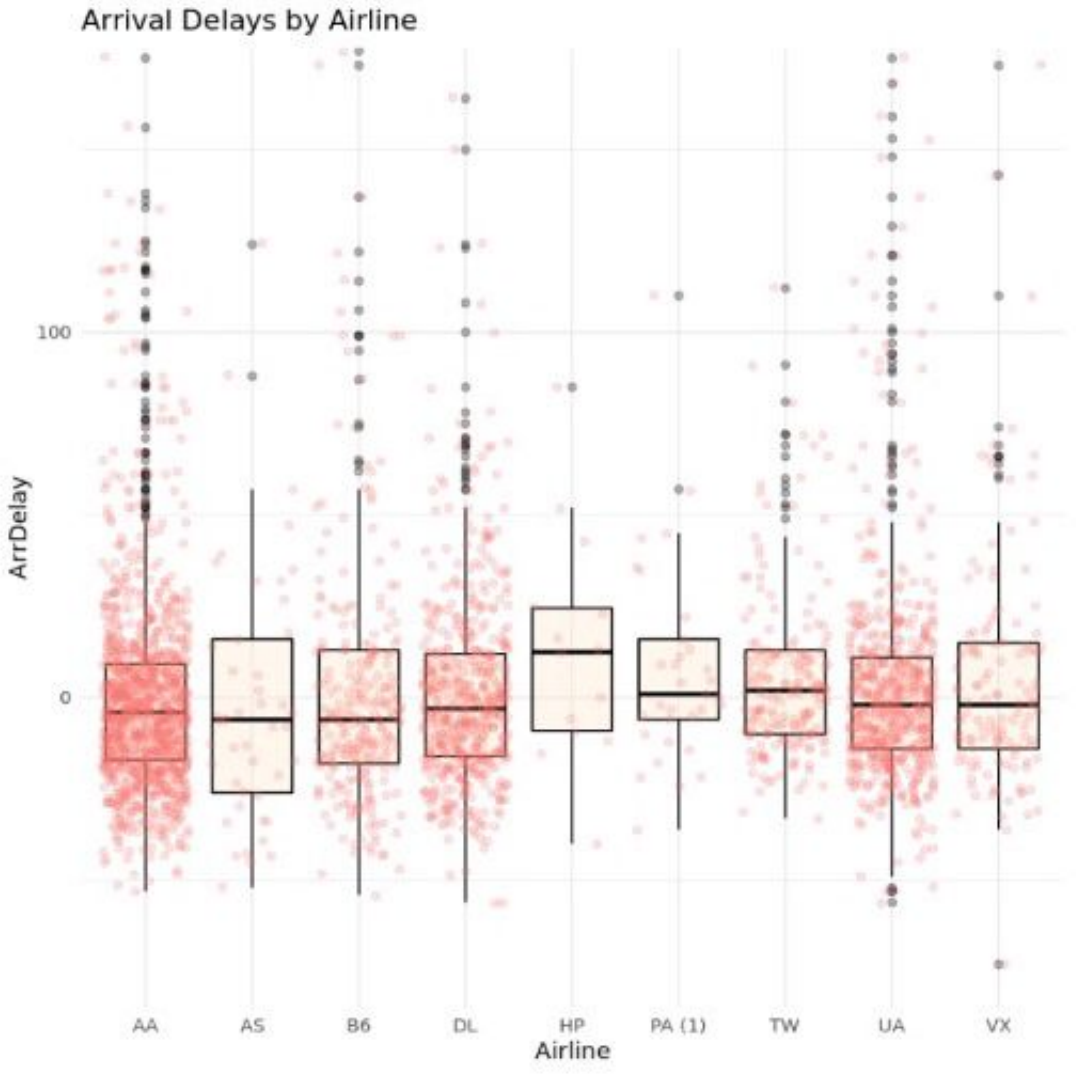

Figura 2 – Atrasos por Companhia Aérea

Fonte: Autoria Própria

A figura 2 mostra o gráfico gerado pelo código acima.

3.4.2 Agrupamento de dados

A fim de analisar a relação entre dados é preciso agrupá-los. Por exemplo: existe alguma relação entre a companhia aérea e o atraso num voo? Se sim o dia da semana tem um impacto nesse tempo? Para responder essas perguntas podemos agrupar os dados referentes a diferentes companhias e comparar os dias da semana.

```
# Media entre Companhias Aereas e Dias da Semana
media_atrasos <- companhias_aereas %>%
    group_by(Reporting_Airline, DayOfWeek) %>%
    summarize(mean_delays = mean(ArrDelayMinutes))
```

```
# Organiza o dataframe, colocando as companhias com maior atraso
media_atrasos %>% arrange(desc(mean_delays))
# Concluimos que PA(1) - Pan American World Airways tem a maior atraso nos
    voos nas Sextas e Sabados
```

3.4.3 Correlação

Trata-se de uma relação matemática entre duas variáveis. Correlação Positiva é quando duas variáveis movem-se na mesma direção, em outras palavras, crescem e diminuem juntas.Correlação Negativa é quando duas variáveis se movem em direções opostas.

O coeficiente de uma correlação varia de -1 (Correlação Negativa Perfeita) a 1 (Correlação Positiva Perfeita). Incluindo o 0 (Não há correlação).

Analisando correlação entre ArrDelayMinutes (Atraso em minutos na chegada) e DepDelayMinutes (Atraso em minutos na partida) obtém-se uma linha linear.

```
ggplot(companhias_aereas, aes(DepDelayMinutes, ArrDelayMinutes)) + geom_
    point() + geom_smooth(method = "lm")
```

Para medir a correlação entre duas variáveis pode-se usar os métodos de Correlação de Pearson: o coeficiente de correlação ou o P-value.

Para interpretar o coeficiente de correlação há as seguintes regras:

Próximo a 1 Correlação Positiva.

Próximo a -1 Correlação Negativa.

Próximo a 0 Não há correlação

Já o P-value possui essas:

P-value $<$ 0.001 Forte certeza no resultado

P-value $<$ 0.005 Moderada certeza no resultado.

P-value < 0.1 Fraca certeza no resultado.

P-value > 0.1 Nenhuma certeza no resultado

Para saber se uma correlação é forte o coeficiente dela deve ser próximo a -1 ou 1. E o P-value deve ser menor que 0.001

```
companhias_aereas %>%
    select(DepDelay, ArrDelay) %>%
# cor eh a funcao que calcula o coeficiente de correlacao
    cor(method = "pearson")
```

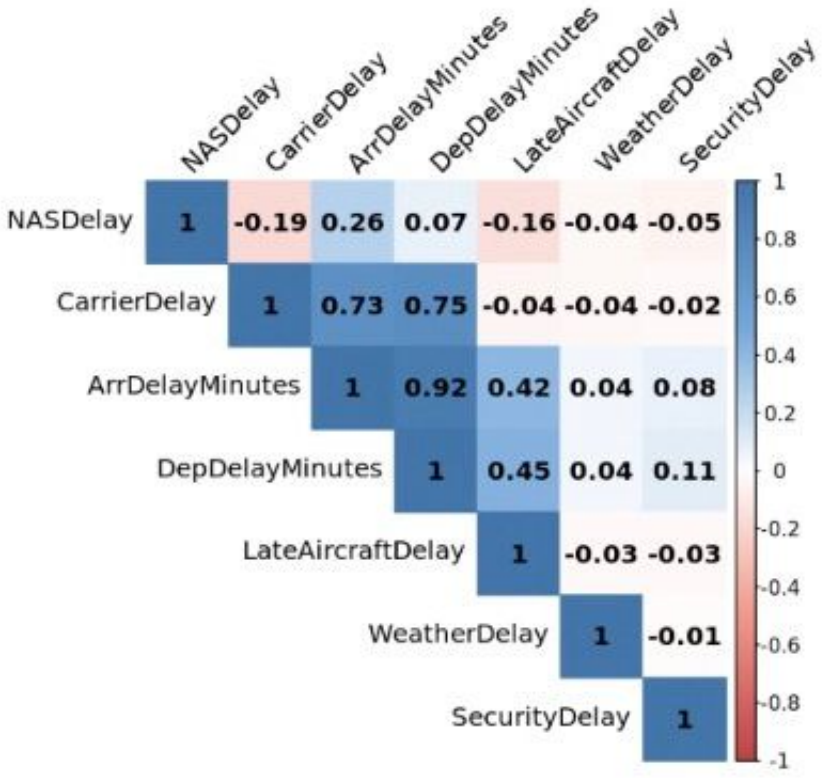

Figura 3 – Correlação entre variáveis

Fonte: Autoria Própria

A figura 3 mostra um gráfico de correlação entre as demais variáveis do dataset. O código para criá-lo é este:

```
install.packages("corrplot")
```

```r
# Vamos ver a correlacao entre mais variaveis
library(corrplot)

valores_numericos <- companhias_aereas %>%
                     select(ArrDelayMinutes, DepDelayMinutes, CarrierDelay,
                          WeatherDelay, NASDelay, SecurityDelay,
                         LateAircraftDelay)

correlacao_companhias <- cor(valores_numericos, method = "pearson", use='
    pairwise.complete.obs')

correlacao_companhias
```

Primeiro calculamos a correlação entre as variáveis ArrDelayMinutes, DepDelayMinutes, CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay para em seguida construir um gráfico.

```r
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "
    #4477AA")) # Cores

corrplot(correlacao_companhias, method = "color", col = col(200),
         type = "upper", order = "hclust",
         addCoef.col = "black", # Adiciona coeficiente de correlacao
         tl.col = "black", tl.srt = 45, # Nome da legenda e sua rotacao
         )
```

3.5 Desenvolvimento de Modelos

Um modelo pode ser visto como uma equação matemática usada para prever um valor dado um ou mais outros valores. Ele relaciona uma ou mais variáveis independentes a variáveis dependentes.

3.5.1 Regressão Linear Simples

Refere-se a uma variável independente para fazer uma predição. É usada para entender a relação entre duas variáveis.

X Variável independente

Y A resposta/variável dependente (o que se quer prever)

Função Linear

$$\hat{Y} = b\_0 + b\_1X$$

b_0 o valor de Y quando X é 0

b_1 coeficiente angular da equação

$\hat{Y}$ o valor previsto pelo modelo

Usando esse modelo faz-se quatro suposições:

Linearidade A relação entre X e a média de Y é linear

Independência Observações são independentes umas das outras

Homoscedasticidade uma sequência de variáveis aleatórias é homocedástica se todas as suas variáveis aleatórias tiverem a mesma variância finita.

Normalidade Para qualquer valor fixo de X, Y é normalmente distribuído

Neste caso, tomando apenas os dados da companhia aérea Alaska Airline(AA) e removendo os valores NA de CarrierDelay(Atrasos causados pela companhia) temos o seguinte modelo.

```
# Definimos um dataset com apenas AA de companhia aerea

alaska <- companhias_aereas %>%
  filter(CarrierDelay != "NA", Reporting_Airline == "AA")

# Mostra os primeiros 6 valores do dataset
head(alaska)
```

Nesse exemplo, usa-se atrasos na partida (DepDelayMinutes) para prever atrasos na chegada (ArrDelayMinutes). Nesse caso, DepDelayMinutes é a variável independente X e ArrDelayMinutes é a variável dependente Y. Usa-se a função *lm()* para criar um modelo linear.

```
# Criamos o modelo
modelo_linear <- lm(ArrDelayMinutes ~ DepDelayMinutes, data = alaska)

# Resumimos o modelo
# b0(Intercept) = 17.35
# b1 = 0.7523
summary(modelo_linear)
```

A equação que obtida para esse modelo é:

$$ArrDelayMinutes = 17.35 + 0.7523 * DepDelayMinutes$$

3.5.2 Regressão Linear Múltipla

Diferente da Regressão Linear Simples, a múltipla refere-se a várias variáveis independentes para fazer uma predição. Ela explica a relação entre uma variável (Y) e duas ou mais variáveis (X).

Equação

$$\hat{Y} = b\_0 + b\_1X\_1 + b\_2X\_2$$

b_0 o valor de Y quando X é 0

b_1 coeficiente da variável 1

b_2 coeficiente da variável 2

Seguindo o exemplo anterior, pode-se usar as variáveis independentes Atraso em minutos da partida (DepDelayMinutes)(Atraso em minutos da partida) e Atraso causado por outro avião (LateAircraftDelay)(Atraso causado por outro avião) para criar outro modelo.

```
regressao_multipla <- lm(ArrDelayMinutes ~ DepDelayMinutes +
    LateAircraftDelay, data = alaska)

summary(regressao_multipla)
```

```
regressao_multipla$coefficients
```

A equação que obtida para esse modelo é:

$$ArrDelayMinutes = 17.32 + 0.7556 * DepDelayMinutes - 0.0103 * LateAircraftDelay$$

3.5.3 Gráficos de Regressão

Fornecem uma boa estimativa:

- ❏ Relação entre duas variáveis
- ❏ A *força* de uma correlação
- ❏ A direção da relação(positiva ou negativa)

São uma combinação de gráficos de dispersão, onde cada ponto representa um Y diferente, e a linha de regressão ($\hat{y}$).

```
# Carregamos a biblioteca ggplot2
library(ggplot2)
# x contem a coluna com a variavel independente e y cont m a dependente.
# Nele temos apenas dados da companhia Alaska
# A funcao geom_point() cria um grafico de dispersao
# A funcao stat_smooth cria a linha vermelha juntamente com a area cinza
    que mostra as margens de erro
ggplot(alaska, aes(x = DepDelayMinutes, y = ArrDelayMinutes)) + geom_point
    () + stat_smooth(method = "lm", col = "red")
```

3.5.4 Erro quadrático médio e coeficiente de determinação

Usa-se essas métricas para validar um modelo.

Para o erro quadrático médio (Erro Quadrático Médio (EQM) ou MSE) é a média dos residuais elevados ao quadrado. Sendo os residuais a diferença entre o melhor ajuste e os dados usados para o treino.

$$EQM = média((\hat{y} - y)^2)$$

O desvio do erro quadrático médio é a raiz do EQM

$$DEQM = \sqrt{EQM}$$

```
# Lembre-se que criamos o modelo_linear anteriormente
EQM <- mean(modelo_linear$residuals^2)
EQM
```

```
# Como a unidade que obtemos desse calculo tambem eh quadratica tiramos a
    raiz.
# A DEQM tera a mesma unidade da variavel Y.
DEQM <- sqrt(EQM)
DEQM
```

O coeficiente de determinação é usado para determinar quão perto o dado está da linha do gráfico de regressão. Na maior parte dos casos ele se encontra entre 0 e 1. O mesmo pode ser obtido a partir do resumo do modelo linear.

```
# Do valor obtido pode-se dizer que o modelo explica 76% da variancia
summary(modelo_linear)$r.squared
```

A fim de determinar se o modelo está correto é preciso saber se os valores previstos por ele fazem sentido.

```
# Se um voo tem um atraso de 12 minutos na partida, o modelo nos preve um atraso na chegada de 26 minutos
# Note que nao obtemos valores discrepantes.

novos_dados <- data.frame(DepDelayMinutes = c(12,19,24))
predicao <- predict(modelo_linear, newdata = novos_dados, interval = 'confidence')
predicao
```

3.6 Análise de Modelos

Em exemplos anteriores foi usado todo o dataset para treinar modelos. Contudo, o objetivo de um modelo é lidar com novos dados e a partir dele fazer previsões. Dessa forma, usando-se os mesmos dados para treino e avaliações o modelo mostrará resultados muito otimistas. Portanto, a ideia é treina-lo com uma parte do dataset e usar a outra para testes.

Nos exemplos a seguir usou-se a biblioteca *tidymodels* para lidar com aprendizado de máquina, construir dados para treino ou teste, etc. Considerando a pergunta mencionada em tópicos anteriores, quais fatores levam ao atraso de um voo. A partir do dataset *companhias_aereas* criou-se o subconjunto atrasos.

```
# Atrasos eh um subconjunto do dataset companhias_aereas.
# Removemos os valores NA
# E selecionamos as colunas que nos interessam (Atrasos causados pelo tempo (WeatherDelay), Mes(Month)...)
atrasos <- companhias_aereas %>%
    replace_na(list(CarrierDelay = 0,
                    WeatherDelay = 0,
                    NASDelay = 0,
                    SecurityDelay = 0,
                    LateAircraftDelay = 0)) %>%
    select(c(ArrDelayMinutes, DepDelayMinutes, CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay, DayOfWeek, Month))
# Imprime o dataframe
atrasos
```

Para construir um modelo é preciso primeiro dividi-lo em um dataset de treino e outro de teste. Então, usa-se a função *set.seed()* para garantir que os subconjuntos gerados a partir do dataset serão sempre os mesmos toda vez que esse código é executado. Observando que por padrão esta função divide 75% dos dados para treino e os 25% restante para teste.

```
set.seed(1234)
```

```
subconjunto <- initial_split(atrasos)
treino <- training(subconjunto)
teste <- testing(subconjunto)
```

Uma vez divido o dataset especifica-se o modelo a ser usado nele. Nesse exemplo, usou-se a regressão linear, a variável atrasos na partida (DepDelayMinutes) para prever atrasos na chegada (ArrDelayMinutes) e treino,por sua vez, é usado para treinar o modelo. O código abaixo demonstra como fazer isso.

```
# Use regressao linear
rl <- linear_reg() %>%
  set_engine(engine = "lm")

# Exiba a funcao
rl
```

```
modelo_ajustado <- rl_modelo %>%
    # Usa-se fit para ajustar o modelo especificado
    fit(ArrDelayMinutes ~ DepDelayMinutes, data = treino)

modelo_ajustado
```

A fim de verificar algumas predições do modelo ajustado usa-se a função *predict()*. Ela produz uma coluna *.pred* com os valores de atrasos na chegada. Nesse caso, ainda não usou-se os dados de teste e sim os de treino. O dataframe criado é salvo na variável resultados_treino.

```
resultados_treino <- modelo_ajustado %>%
  # Faca previsoes e salve os valores
  predict(new_data = treino) %>%
  # Crie uma nova coluna originais com do treino
  mutate(originais = treino$ArrDelayMinutes)

head(resultados_treino)
```

Para testar o modelo só é preciso trocar a variável treino por teste.

```
resultados_teste <- modelo_ajustado %>%
  predict(new_data = teste) %>%
  mutate(originais = teste$ArrDelayMinutes)

head(resultados_teste)
```

Para verificar a validade desse modelo, pode-se usar as métricas EQM ou o coeficiente de determinação mencionados anteriormente. Ao invés de aplicar manualmente essas fórmulas é melhor usar as funções *rmse()* e *rsq()*

```
# Calcula a raiz do erro quadratrico medio dos dados obtidos a partir do
    treino
rmse(resultados_treino, truth = originais,
     estimate = .pred)
```

```
# Calcula a raiz do erro quadratrico medio dos dados obtidos a partir do
    teste
rmse(resultados_teste, truth = originais,
     estimate = .pred)
```

```
# Calcula o coeficiente de determinacao dos dados obtidos a partir do
    treino
rsq(resultados_treino, truth = originais,
    estimate = .pred)
```

```
# Calcula o coeficiente de determinacao dos dados obtidos a partir do teste
rsq(resultados_teste, truth = originais,
    estimate = .pred)
```

Capítulo 4

Conclusão

O propósito desse trabalho foi introduzir o leitor ao estudo da Ciência de Dados. Inicialmente, portanto, foram explicadas as vantagens de ser um profissional da área ("Data Scientist:The Sexiest Job of the 21st Century"), o impacto que ela terá na sociedade e o cenário atual, levando em conta a implementação da Lei Geral de Proteção aos Dados (LGPD). Como mencionado, Amaral vê Big Data como a 4° revolução industrial.

Em seguida, o leitor conheceu o Ciclo de Vida do Dado, aprendendo sobre o processo de coleta, análise e descarte.Por último no capítulo 3, o uso da linguagem R no estudo de dados. Nesse sentido, foram apresentados exemplos usando o dataset Airline da empresa IBM. O problema, então apresentado era como determinar os fatores que levam ao atraso de um voo. Por exemplo: atrasos causados pelo clima, segurança, equipe técnica e outros.

A partir deles, construiu-se o dataset *companhias_aereas*. Com ele o leitor aprendeu a fazer uma limpeza e processamento de dados. Remover valores NA e fazer a normalização deles, por exemplo. Depois disso, foram abordadas técnicas usadas em análises exploratórias e desenvolvimento de modelos, dentre elas o uso de Regressão Linear, a construção de modelos matemáticos, agrupamento de dados e outras.

Ao final deste trabalho, o leitor é capaz de usar ferramentas como JupyterLab e outros ambientes de programação que utilizem a linguagem R. Também ganhou noções básicas de análise de dados e aprendizado de máquina usadas para realizar previsões a partir de um dataset dado, por exemplo, qual linha aérea tem o menor tempo de atraso em voos de Los Angeles a Nova York. Concluindo, conheceu melhor a área de Ciência de Dados.

Este trabalho não cobre, é claro, o escopo completo da Ciência de Dados. Tópicos como Árvore de Decisão, Mineração de Texto e Grafos foram deixados de lado em favor de técnicas mais básicas. Resgatando a ideia da alta demanda por profissionais nessa área, abordada no capítulo introdutório, é recomendado ao leitor ir além do que foi apresentado aqui, seja fazendo cursos na plataforma Coursera ou lendo mais pesquisas como esta.

Referências

AMARAL, F. **Introdução à Ciência de Dados**. Rio de Janeiro, RJ, Brasil: ALTA BOOKS, 2016.

DAVENPORT, T. H.; PATIL, D. J. Data scientist: The sexiest job of the 21st century. **Havard Business Review**, 2012.

LGPD Lei Geral de Proteção de Dados Pessoais. Brasília, DF, Brasil: Secretaria-Geral Subchefia para Assuntos Jurídicos, 2018.