UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL TEIXEIRA CASCHERA

**CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS***

São Carlos

2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL TEIXEIRA CASCHERA

**CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS***

Monografia apresentada ao Departamento de Computação da Universidade Federal de São Carlos para obtenção do título de bacharel em Engenharia de Computação.

Orientação: Prof. Dr. Diego Furtado Silva

São Carlos

2019

GABRIEL TEIXEIRA CASCHERA

CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS*

Monografia apresentada ao Departamento de Computação da Universidade Federal de São Carlos, para obtenção do título de bacharel em Engenharia de Computação. Universidade Federal de São Carlos. São Carlos, XX de julho de 2019.

Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Diego Furtado Silva

Departamento de Computação - UFSCar

Examinador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

XXXX

Departamento de Computação - UFSCar

Examinador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

XXXX

Departamento de Computação - UFSCar

**DEDICATÓRIA**

*XXXXX.*

**AGRADECIMENTO**

XXXXX

**RESUMO**

XXXXX

Palavras-chave: X1.X2.

**ABSTRACT**

XXXXXX

Keywords: X1. X2.

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc11532851)

[2 ANÁLISE EXPLORATÓRIA DE DADOS 9](#_Toc11532852)

[3 LEVANTAMENTO DE FERRAMENTAS PARA ANÁLISE DE DATASETS 10](#_Toc11532853)

[4 VISUALIZAÇÕES GRÁFICAS A SEREM CONTEMPLADAS NA FERRAMENTA 12](#_Toc11532854)

[4.1 Conceitos estatísticos básicos 12](#_Toc11532855)

[4.2 *Box plot* 15](#_Toc11532856)

[4.3 Histograma 15](#_Toc11532857)

[4.4 Histogramas 15](#_Toc11532858)

[4.5 Gráficos de Dispersão (*scatterplots*) 15](#_Toc11532859)

[5 ESCOLHA DE LINGUAGEM DE PROGRAMAÇÃO PARA DESENVOLVIMENTO 15](#_Toc11532860)

[6 LEVANTAMENTO DE BIBLIOTECAS DE VISUALIZAÇÃO DE DADOS NO PYTHON 16](#_Toc11532861)

[6.1 Matplotlib 16](#_Toc11532862)

[6.2 Seaborn 17](#_Toc11532863)

[6.3 ggplot 17](#_Toc11532864)

[6.4 Bokeh 17](#_Toc11532865)

[6.5 Plotly 17](#_Toc11532866)

[6.6 Consideração final 18](#_Toc11532867)

[7 CONSTRUÇÃO DA FERRAMENTA 18](#_Toc11532868)

[8 RESULTADOS 19](#_Toc11532869)

[9 CONCLUSÕES 19](#_Toc11532870)

[10 BIBLIOGRAFIA 20](#_Toc11532871)

1. INTRODUÇÃO

Nos últimos 20 anos, a quantidade de dados coletados dentro das empresas, em todas as frentes (operações, campanhas de *marketing*, manufatura, entre outros), tornou-se cada vez maior. Além disso, dados externos às organizações também tornaram-se cada vez mais fáceis de serem acessados. Esses fatores, combinados, permitiram o crescimento do interesse na criação de métodos para a extração de informações úteis e conhecimento a partir desses dados. (*Provost, F.;Fawcett, T.; Data Science for Business – O’Reilly, 2013*)

Devido a esse interesse, o número de aplicações na área de *Data Science* cresceu. A correta análise e preparação dos dados é muito importante para estas aplicações e, com o crescimento da utilização e da disponibilidade de dados, essa etapa torna-se cada vez mais importante. Grande parte do tempo de construção e teste de modelos estatísticos para essas ferramentas de *Machine Learning* é gasto justamente nesta etapa de entendimento dos dados.

Em casos de projetos cujos dados não possuem tanta qualidade, isto é, estão em desacordo com o fenômeno a ser estudado/previsto ou estão faltando informações, é de suma importância reconhecer esses detalhes logo no início do projeto. Por exemplo, ao se construir um modelo para seleção de um canal de cobrança para cada cliente de uma base de dados, o primeiro passo é entender se os dados de fato permitem que esse modelo seja aplicado.

Atualmente, a maior parte das empresas utilizam o Microsft Office Excel como principal ferramenta para a análise de dados. Apesar de ser uma ferramenta bastante poderosa para esse contexto, está longe de ser capaz de processar grandes quantidades de dados e construir modelagens robustas, devido às suas limitações de performance e compatibilidade.

Deste modo, é proposta uma ferramenta para facilitar o rápido diagnóstico de *datasets*, através da apresentação de dados básicos do *dataset* e da fácil criação de visualizações de informações gráficas pelo usuário.

1. ANÁLISE EXPLORATÓRIA DE DADOS

O termo Análise Exploratória de Dados (em inglês, *Exploratory Data Analysis – EDA*) refere-se ao entendimento das características e geração de inteligência e *insights* com base na exploração dos dados disponíveis. Isto é, através de análises estatísticas e visualização das variáveis presentes nos conjuntos de dados, gerar valor e entendimento do fenômeno que gerou estes dados. (Tukey 1977)

A realização da EDA traz uma série de benefícios em relação ao entendimento dos dados disponíveis: detecção de erros nos *datasets*, validação de hipóteses, escolha do modelo estatístico a ser aplicado posteriormente e entendimento das variáveis de *input* e *output*. (*Seltman HJ 2012*) Assim, é notável a importância da realização da EDA para a aplicação de técnicas de *machine learning*, já que a mesma traz o entendimento necessário para que seja possível escolher entre os diversos modelos disponíveis aquele que melhor atende ao caso com os dados disponíveis.

Segundo matéria da revista Forbes (referência Forbes), a quantidade de dados produzida nunca foi tão grande, chegando a 2.5 quintilhões bytes de dados gerados todos os dias. Esse número é tão grande especialmente por conta da *Internet of Things* (IoT, internet das coisas), mas outros contribuintes são serviços como previsão do tempo e entretenimento, as redes sociais e os meios de comunicação online.

Deste modo, os dados coletados e disponibilizados hoje podem ser trabalhados e entendidos através da EDA, levando a melhores escolhas de modelos de *machine learning*, isto é, modelos que possam gerar bons resultados a partir dos dados de *input* fornecidos, e gerando inteligência e *insights*.

Dentro do mundo dos negócios, são citados dois casos: o primeiro, da rede Wal-Mart, um dos maiores grupos de varejo do mundo, conta como esta empresa se beneficiou de seus dados para planejar-se para um momento difícil; o segundo, ...

1. LEVANTAMENTO DE FERRAMENTAS PARA ANÁLISE DE DATASETS

Foi levantada uma ferramenta disponível para a realização de análise exploratória de dados: DataExplore, construída pelo Dr. Damien Farrell (2016). Além disso, é realizada uma comparação com o *software* Microsoft Office Excel, comumente utilizado no ambiente corporativo como ferramenta para realizar análises exploratórias rápidas.

A ferramenta DataExplore foi proposta devido à necessidade, segundo o autor, de uma forma mais simples de analisar os dados dentro da ciência. Atualmente, as duas formas mais utilizadas são o Excel e os *scripts* criados através de linguagens de programação, tais quais R e Python.

A primeira opção permite que o usuário analise e gere visualizações gráficas, mas apresenta limitações de capacidade, como número de linhas e opções limitadas de funcionalidades, e dificuldades de implementação, como o cruzamento de bases, em relação à segunda. Além disso, não fornece uma forma segura de análise dos dados, já que esta tarefa ocorre no mesmo ambiente que os dados originais são armazenados, possibilitando mudanças na informação original.

Quanto à segunda opção citada pelo autor, as linguagens de programação são, para muitos usuários, difíceis de compreensão e utilização, desfavorecendo a utilização das mesmas. Assim, por mais que sejam uma opção que oferece maleabilidade de uso, pois permitem que o usuário controle por inteiro tanto a análise quanto a geração de visualizações, acabam não sendo escolhidas por alguns usuários.

Deste modo, Dr. Damien Farrell constrói uma ferramenta *open-source* que permite o cruzamento de bases de dados, gerando novas visões das informações, e a geração de visualizações gráficas de forma simples, sem exigir que o usuário tenha habilidades de programação. Ainda, o DataExplore é capaz de armazenar como as visualizações foram geradas, para que seja possível replicar uma mesma tarefa no futuro.

Tais características do DataExplore tornam esta ferramenta muito interessante para utilização, especialmente por usuários que não conhecem tanto de programação e que estão em busca de uma forma de gerar visualizações gráficas de qualidade com facilidade.

Além dos pontos negativos do *software* Excel citados anteriormente (limitações de uso e dificuldades para implementação), esta ferramenta possui um custo de utilização (o que torna soluções *open-source*, como o DataExplore, mais atrativas). Estas características fazem com que a mesma não seja a melhor opção para análise exploratória de dados, mesmo que seja uma das mais utilizadas no mercado.

Ferramentas corporativas de visualização de dados como Tableau[[1]](#footnote-1) e Power BI[[2]](#footnote-2) também podem ser utilizadas para o fim de análise exploratória de dados. Elas oferecerem meios para realizar cruzamento de bases e adicionar colunas calculadas, por exemplo, tendo como *inputs datasets* em bases do SQL Server, arquivos *.xlsm* do Excel, *.accdb* do Microsoft Access, entre outros. Entretanto, devido às suas funcionalidades principais serem a criação de *dashboards online* (i.e., painéis para acompanhamento de dados em tempo real), optou-se por não considerar as mesmas como opções para comparação neste trabalho.

1. VISUALIZAÇÕES GRÁFICAS A SEREM CONTEMPLADAS NA FERRAMENTA

Para atingir o objetivo da ferramenta de permitir realizar análise exploratória dos dados e entender melhor o comportamento das variáveis no *dataset*, foram escolhidas e revisadas visualizações e funções estatísticas.

Os dados utilizados para geração dos exemplos apresentados nas próximas seções foram obtidos através da plataforma Kaggle[[3]](#footnote-3). Nela, é possível, entre outras coisas, enxergar uma prévia dos dados, o conjunto de meta-dados e descrições das bases que são disponibilizadas, além de ser possível realizar *download* dos *datasets* em csv (*comma separeted values*).

A geração dos gráficos apresentados como exemplos foi realizada utilizando a linguagem de programação Python. A biblioteca para leitura e organização dos dados foi a Pandas, enquanto que aquela utilizada para construção das visualizações foi a Seaborn.

* 1. CONCEITOS ESTATÍSTICOS BÁSICOS

Para o entendimento das visualizações e das informações dadas através das mesmas, primeiro faz-se necessário entender os principais conceitos estatísticos envolvidos. Somente através dos mesmos é possível compreender o comportamento das variáveis presentes ou derivadas do conjunto de dados em análise e interpretar corretamente as visualizações gráficas.

Essencialmente, existem dois tipos de dados estatísticos: os numéricos (ou quantitativos) e os categóricos (ou qualitativos). Os primeiros podem ser medidos de forma contínua, como a velocidade de um carro ao longo do tempo ou a frequência de transmissão de dados através de uma conexão de rede, ou discreta, como o número de ocorrências de tornados no sul da Flórida (EUA) entre os anos 2000 e 2010 ou a quantidade de vendas realizadas via *e-commerce* no Brasil em 2018.

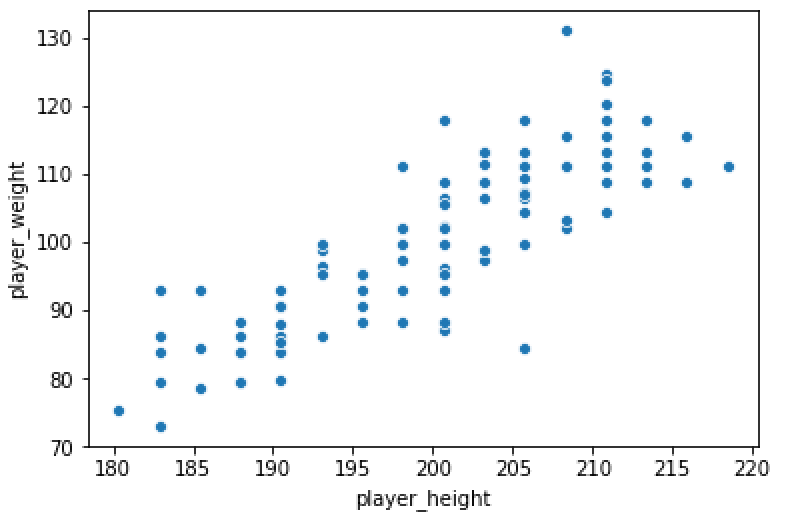
Os segundos representam classes e possuem um conjunto finito de possibilidades a serem assumidas. Essas categorias podem ser os estados brasileiros (“SP”, ”RJ”, “ES”, etc.) ou o canal de vendas por onde um dado cliente realizou sua compra (“loja física”, “*e-commerce*”, “televendas”). A variável categórica pode ser binária, indicando se um dado elemento pertence ou não à classe representada, como no caso de uma variável que indica se o usuário é ou não administrador de um dado sistema (“1” ou “0”, “sim” ou “não”, “*true*” ou “*false*”). Ainda, pode ser ordinal, indicando ordem entre os elementos, como a colocação de pilotos de Fórmula 1 após uma corrida.

Para melhor entender o comportamento das variáveis numéricas e o que elas representam, são utilizadas medidas estatísticas como a média, mediana e a moda, entre outros. Cada uma delas possui uma sensibilidade diferente aos dados e fornecem informações sobre a distribuição dos mesmos.

A média representa a soma de todos os valores do conjunto dividido pelo número de elementos no mesmo.

(1)

Esta medida é muito suscetível a *outliers*, i.e., amostras que fogem ao padrão apresentado pelo conjuntado de dados. Um exemplo de *outlier* é apresentado, marcado em vermelho, no Gráfico 1. O jogador representado pelo ponto foge aos padrões apresentados pelos outros, já que seu peso é muito menor do que aquele apresentado pelos outros jogadores da sua altura. É muito importante distinguir *outliers* de ruídos nas amostras. Diferente de ruídos, que fogem ao padrão por conterem erros na obtenção dos dados (por exemplo, um mau funcionamento num dado sensor), *outliers* são medidas corretas que fogem às características padrões do grupo.

**GRÁFICO 1** – Altura x Peso de jogadores da NBA com mais de 30 anos na temporada 2016-17 

Fonte: <https://www.kaggle.com/justinas/nba-players-data>

Outra medida comum para essas análises é a mediana. Ela consiste no valor da amostra no meio de um conjunto ordenado, isto é, amostra a qual 50% das outras amostras estão abaixo dela (num conjunto ordenado). Por exemplo, numa sala de aula com 9 (nove) alunos, com suas respectivas notas apresentadas na Tabela 1, a mediana será a nota do quinto aluno, ou seja, 7,5, pois 50% das amostras estão abaixo deste. Caso o conjunto de dados apresente número par de elementos, considera-se a média entre o último elemento da primeira metade e o primeiro elemento da segunda metade.

**TABELA 1** – Exemplo de notas de 9 alunos

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aluno | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Nota | 5,0 | 5,2 | 6,3 | 7,0 | 7,5 | 7,7 | 8,2 | 8,3 | 8,5 |

A mediana não é sujeita à influência dos *outliers*, diferentemente da média, pois ela não leva em consideração o valor de todas as amostras, somente sua posição no conjunto. Utilizando o exemplo das notas dos alunos, caso o último aluno tivesse obtido uma nota 10 ao invés de 8,5 e o primeiro uma nota 0 ao invés de 5, a mediana permaneceria a mesma.

A mediana é coincidente com o segundo quartil, que representa o percentil 50%. Percentil 50% significa que 50% das amostras possuem valor abaixo daquele (e, consequentemente, que os outros 50% possuem valor acima). De maneira geral, pode-se dizer que um percentil x% representa o valor que x% das amostras estão abaixo dele e os outros (100-x%) estão abaixo. Deste modo, os primeiro, segundo e terceiro quartis são representados, respectivamente, pelos percentis 25%, 50% e 75%. Define-se a distância interquartis como sendo a diferença entre o terceiro e o primeiro quartis.

Por fim, as visualizações e análises de dados podem ser divididas em dois tipos: univariada e multivariada. O primeiro trata de atributos que variam com relação a somente um outro atributo, enquanto o segundo trata de atributos que variam conforme duas ou mais outras variáveis.

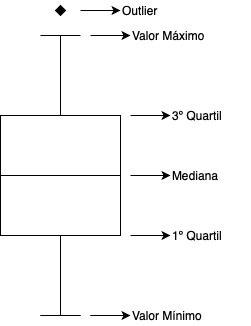
Dadas estas definições, são apresentadas algumas formas de visualizações de dados ao longo das próximas subseções.

* 1. *BOX PLOT*

O *box plot* mostra a distribuição das amostras através de suas medidas de mediana, primeiro e terceiro quartis, além de mostrar os valores máximo e mínimo do conjunto, bem como *outliers*. A Figura 1 explica estes elementos de forma visual.

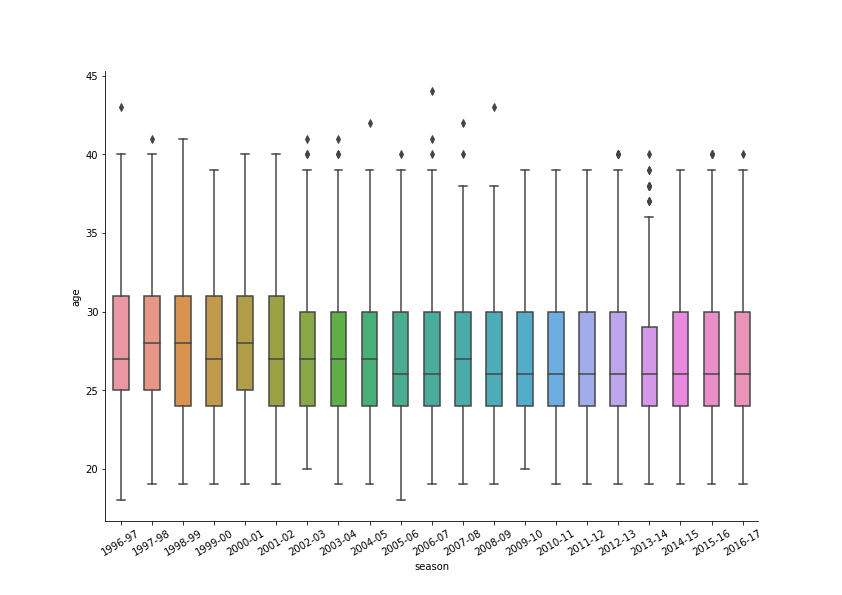
Com esta visualização, é possível enxergar a distribuição do conjunto, analisando quão compacto o mesmo se encontra e se é um conjunto simétrico, permitindo tirar conclusões sobre os dados serem ou não enviesados.

**FIGURA 1** – Explicação sobre o box plot



O Gráfico 2 mostra a distribuição das idades dos jogadores da NBA ao longo das temporadas utilizando *box plots*. A partir destes, pode-se ver que está é uma variável praticamente simétrica para quase todas as temporadas do esporte, com destaque para a temporada de 2007-08. Ainda, pode-se inferir que a idade máxima dos jogadores caiu ao longo das temporadas.

**GRÁFICO 2** – Distribuição das idades dos jogadores da NBA ao longo das temporadas – 1996 a 2016

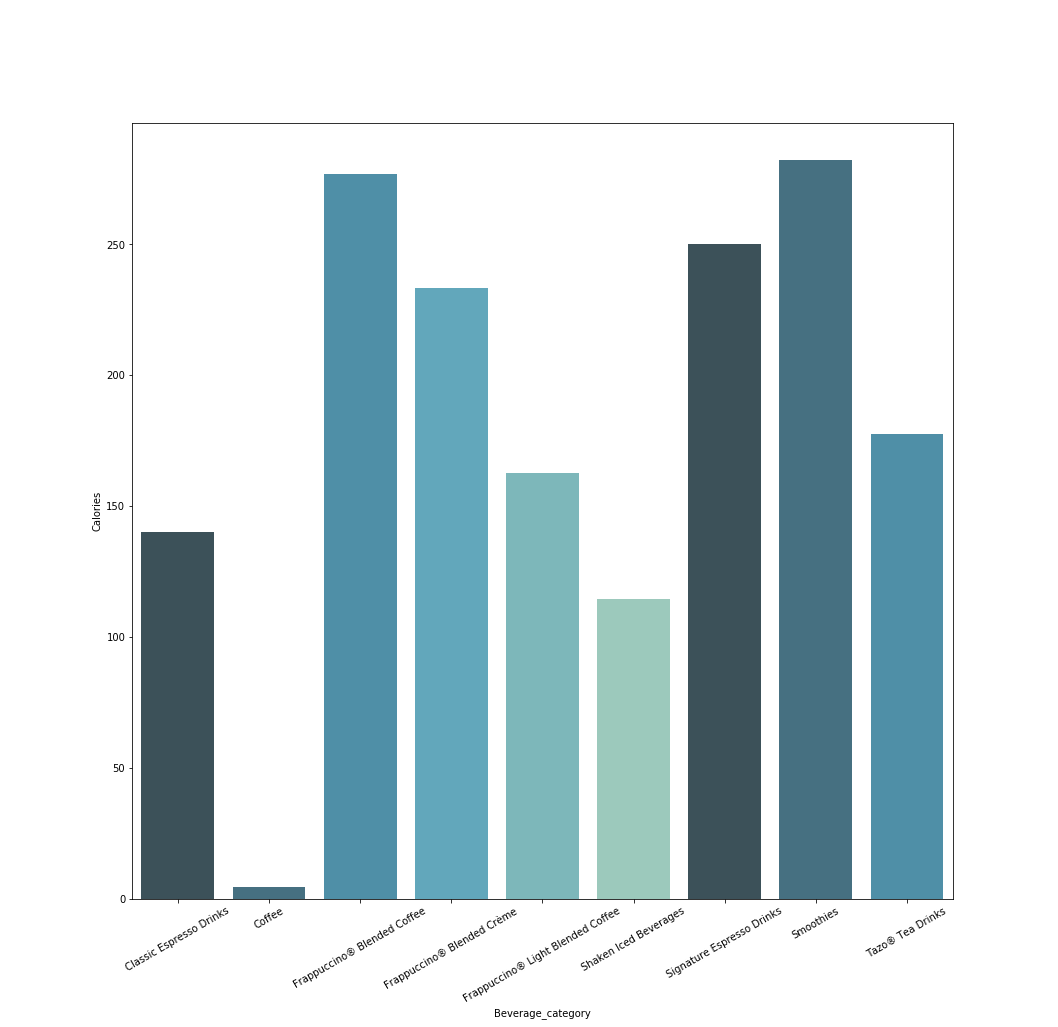


Fonte: <https://www.kaggle.com/justinas/nba-players-data>

* 1. GRÁFICOS DE COLUNA E HISTOGRAMAS

As visualizações apresentadas em formato de coluna são utilizadas para comparação de dados contínuos sobre dados discretos. Desta forma, é possível comparar categorias, já que o eixo y deste tipo de gráfico apresenta o valor resultante do agrupamento das categorias presentes no eixo x. O Gráfico 2 mostra a quantidade média de calorias por bebida para cada um dos grupos de bebidas da rede Starbucks. Neste caso, o eixo y representa a média do agrupamento dos dados pertencentes a cada uma das nove categorias presentes no eixo x. Através desta visualização, pode-se ver que bebidas na categoria de café (*Coffee*), são as menos calóricas, enquanto bebidas mais sofisticadas, como *Smoothies*, apresentam mais calorias por porção.

**GRÁFICO 3** - Média de calorias por grupo de bebidas da rede Starbucks

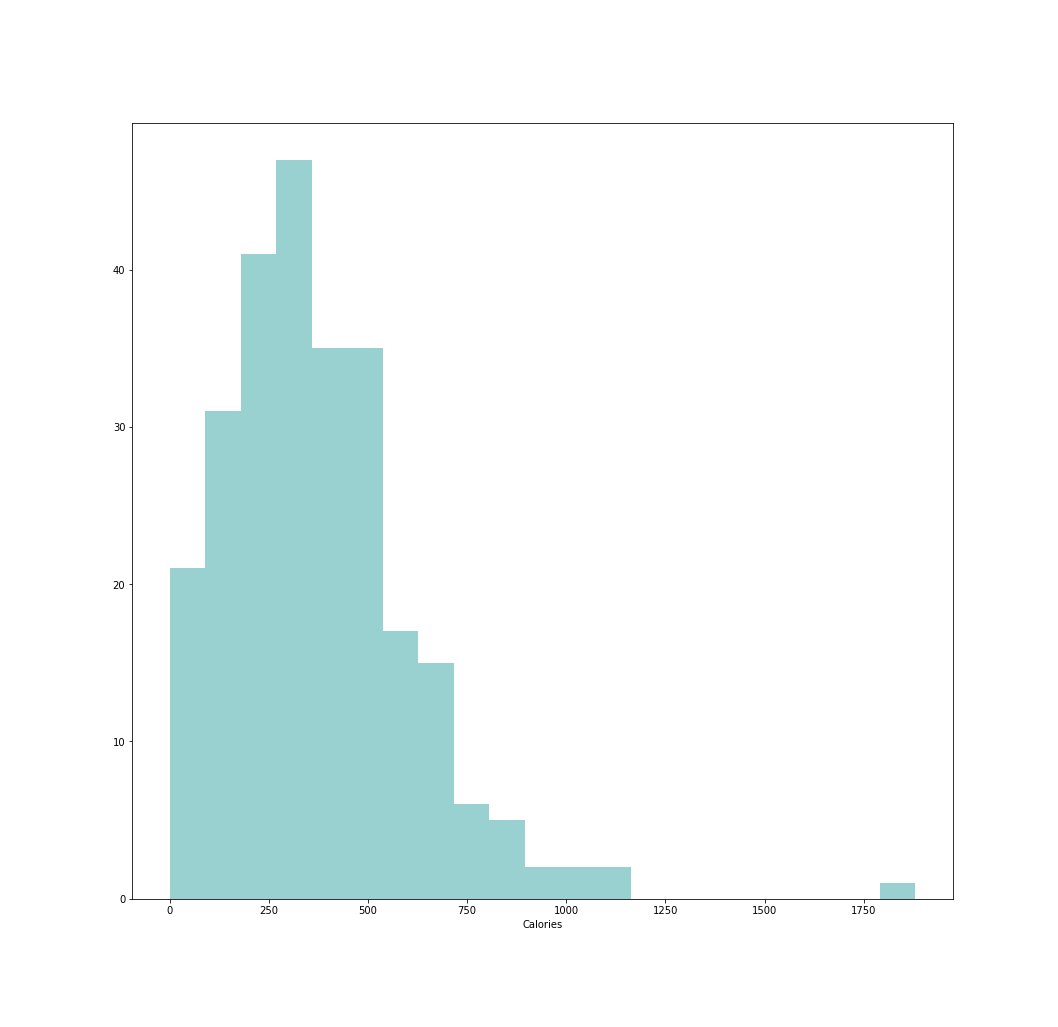


Fonte: <https://www.kaggle.com/starbucks/starbucks-menu/>

Outro uso muito comum de gráficos de barra é a construção de histogramas. Esta visualização consiste no número de ocorrências de um dado valor presente no conjunto de dados. Através dos histogramas, é possível enxergar como ocorre a distribuição de frequência das amostras acerca de uma dada característica do *dataset*. Logo, é possível entender se uma variável possui seus valores distribuídos de forma balanceada ou não, o que pode influenciar tanto as métricas calculadas no conjunto (média e mediana, por exemplo), assim como decisões tomadas a partir desses dados.

O Gráfico 4 representa um histograma que mostra a quantidade de itens do menu da rede americana de *fast food* McDonald’s por quantidade de calorias presentes nos mesmos. Deste forma, enxerga-se a quantidade de itens para cada faixa de calorias. A maior parte dos itens possui entre 250 e 375 calorias, sendo que alguns poucos possuem mais de 1000 calorias. A média e a mediana são, respectivamente, 368,27 e 340 calorias.

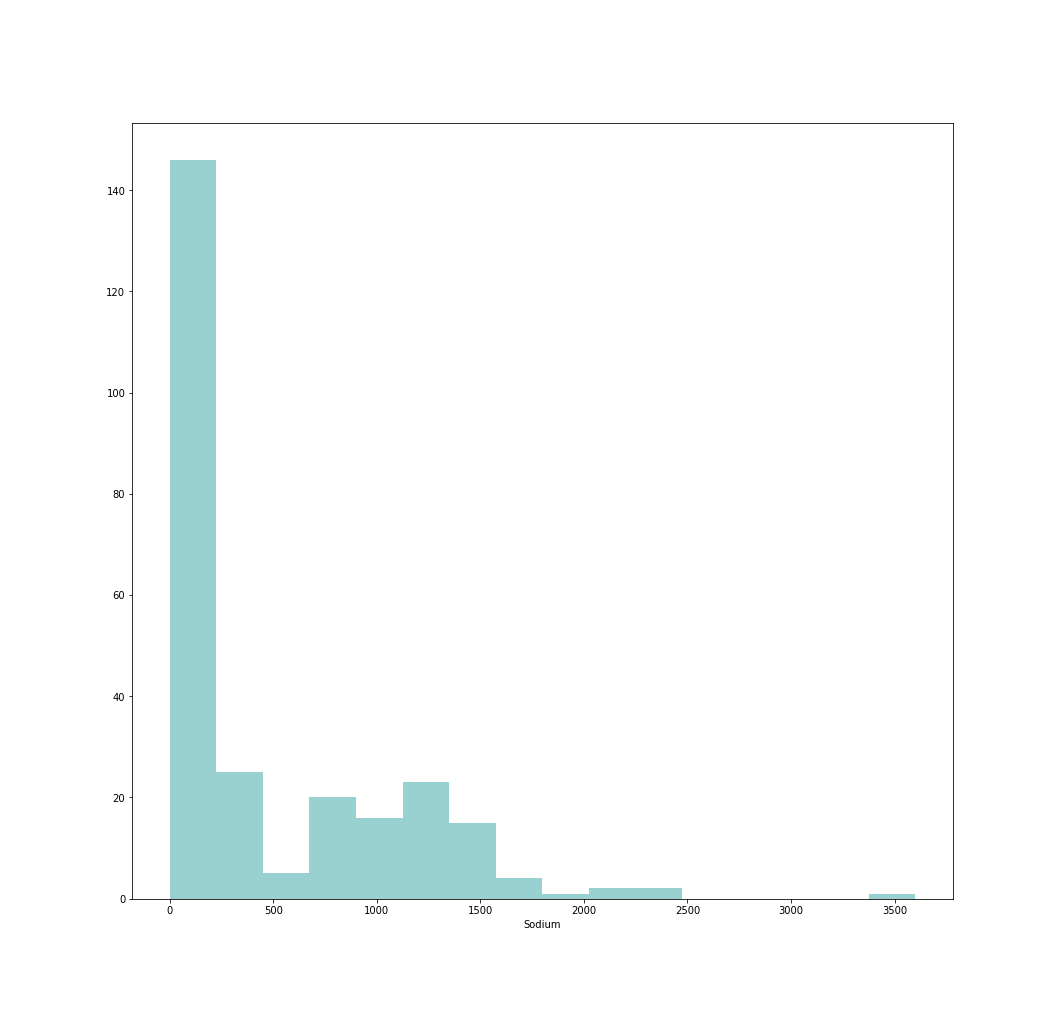
**GRÁFICO 4** - Quantidade de itens do menu da rede McDonald’s por quantidade de calorias



Fonte: ﻿<https://www.kaggle.com/mcdonalds/nutrition-facts/>

Por outro lado, a distribuição da quantidade de itens por quantidade de sódio não apresenta o mesmo comportamento, como mostra o Gráfico 5. Existem muitos itens com menos de 250mg de sódio, mas também existem muitos itens acima desta quantidade, sendo que alguns deles apresentam valores acima de 2000. Os valores de média e mediana comprovam esta distribuição não uniforme. Seus valores são, respectivamente, 495,75 e 190. Este também é um exemplo de como a média é mais influenciada por *outliers* que a mediana.

**GRÁFICO 5** - Quantidade de itens do menu da rede McDonald’s por quantidade de sódio



Fonte: ﻿<https://www.kaggle.com/mcdonalds/nutrition-facts/>

* 1. GRÁFICOS DE LINHA

[Definições]

[Rodar e analisar exemplos]

* 1. GRÁFICOS DE DISPERSÃO (*SCATTERPLOTS*)

[Definições]

[Rodar e analisar exemplos]

* 1. GRÁFICOS DE CALOR (*HEATMAPS*)

[Definições]

[Rodar e analisar exemplos]

* 1. GRÁFICOS DE VIOLINO (*VIOLIN PLOTS*)

[Definições]

[Rodar e analisar exemplos]

1. ESCOLHA DE LINGUAGEM DE PROGRAMAÇÃO PARA DESENVOLVIMENTO

Para o desenvolvimento da ferramenta proposta, foi escolhida a linguagem de programação Python. Essa linguagem possui alta compatibilidade com fontes de dados, a exemplo do SQL Server, através da biblioteca *pyodbc*[[4]](#footnote-4), de arquivos *csv* (*comma separated* values) e do Excel, ambos com funções da Pandas[[5]](#footnote-5).

Segundo o índice TIOBE de maio de 2019 [REF], realizado pela empresa holandesa homônima focada em qualidade de códigos, Python é a quarta linguagem de programação mais popular em 2018 e 2019, ficando atrás de Java, C e C++. Este índice é calculado com base no número de pesquisas pelas linguagens de programação em ferramentas de busca, como Google, e outras fontes de informação, como a Amazon e a Wikipedia.

Entretanto, de acordo com pesquisa realizada pelo *site* KDnuggets (Gregory Piatetsky-Shapiro, Ph.D) [REF], o Python foi elegido como a linguagem mais popular para projetos de *machine learning* e *big data*. A pesquisa levou em conta a opinião de 1800 participantes, que escolheram as ferramentas que utilizavam para projetos na área de *Data Science*. As opções dadas para os participantes contemplavam, entre outras, Python, R e SQL. Um terço dos usuários que votaram em somente uma ferramenta foram removidos, de modo a prevalecer a opinião de usuários ativos dentro da área de estudo em questão.

Deste modo, devido à sua popularidade, facilidade de aprendizado [REF], implementação e codificação, alta disponibilidade de bibliotecas e pacotes e compatibilidade com diversas fontes de dados e ferramentas, Python foi escolhida para a implementação deste trabalho.

1. LEVANTAMENTO DE BIBLIOTECAS DE VISUALIZAÇÃO DE DADOS NO PYTHON

Com a escolha do Python como linguagem de programação para implementação das funções da ferramenta, é necessário encontrar um biblioteca que seja capaz de fornecer as visualizações gráficas necessárias (descritas na Seção Para realização das visualizações, foram levantadas algumas bibliotecas do Python e exploradas suas principais funcionalidades e características.

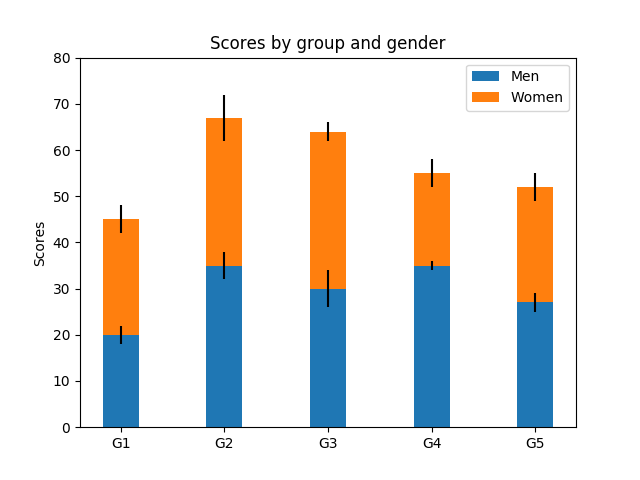
A seguir, são apresentados o que foi encontrado em cada uma das bibliotecas e indicada a escolha para construção da ferramenta.

* 1. MATPLOTLIB

Primeira biblioteca dedicada à visualização de dados no Python, pode ser utilizada para a geração de vários tipos de gráficos, desde histogramas até gráficos de linha. Consegue ser utilizada para criação de gráficos com poucas linhas de código, facilitando muito o desenvolvimento de *software*. Suas visualizações não são dinâmicas, o que dificulta a simples interação do usuário com os gráficos gerados.

A Figura 2 mostra um exemplo de gráfico de barras empilhadas gerado utilizando-se a Matplotlib e disponibilizado na galeria *online* desta biblioteca. Nesta fonte, também é possível encontrar o código Python para gerar essa visualização. HUNTER, J. et al. **Matplotlib Homepage**. Disponível em: <https://matplotlib.org/index.html>. Acesso em: 18 maio. 2019.

**FIGURA 2 -** Exemplo de gráfico de barras empilhadas gerado pela Matplotlib

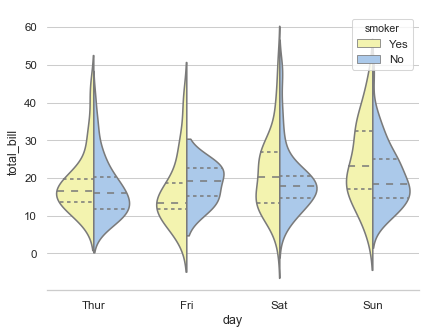


Fonte: <https://matplotlib.org/gallery/lines_bars_and_markers/bar_stacked.html#sphx-glr-gallery-lines-bars-and-markers-bar-stacked-py>

* 1. SEABORN

Construída em cima do Matplotlib, apresenta maior facilidade para utilização de *plots* mais complexos, como mapas de calor e gráficos violinos, por ser uma biblioteca mais alto nível. Além disso, suas cores e apresentações são consideradas mais bonitas que aquelas da biblioteca anterior. Como o Matplotlib, suas visualizações são estáticas. A Figura 3 apresenta exemplo de um gráfico de violino (*violin plot*), dado como exemplo na galeria *online* da Seaborn.

**FIGURA 3** - Exemplo de gráfico gerado pela biblioteca Seaborn



Fonte: <https://seaborn.pydata.org/examples/grouped_violinplots.html>

* 1. GGPLOT

Baseado nas ideias do livro *The Grammar of Graphics* e na biblioteca *ggplot2* do R, o *ggplot* permite a criação de gráficos menos personalizados, focando na reutilização de código e aceleração da geração dos gráficos.

Esta biblioteca conversa muito com a biblioteca de processamento de dados *Pandas* e seu modo de guardar os dados em *dataframes*. Sua implementação faz com que seja possível criar gráficos em camadas, adicionando *plots* um em cima do outro.

* 1. BOKEH

Implementada com base no *The Grammar of Graphics*, como o *ggplot*, é uma biblioteca nativa do Python. Permite a criação de gráficos interativos, com ferramentas de *zoom*, movimentação, atualização dos dados e opção de salvar a visualização. Permite que os *plots* sejam exportados em JSON, documentos HTML ou em interfaces *web*. Os dados a serem apresentados podem ser fornecidos em listas do próprio Python ou estruturas de dados do *NumPy* e *Pandas*.

* 1. PLOTLY

Esta biblioteca é construída em cima da *D3.js*, que é uma biblioteca JavaScript implementada para conectar quaisquer dados a um *Document Object Model* (DOM), o qual, por sua vez, pode ser transformado conforme o necessário, gerando vários tipos de visualizações. Deste modo, ela é capaz de, com poucas linhas de código, gerar visualizações gráficas completas, com as quais o usuário pode interagir: aplicar e tirar *zoom*, navegar pelo gráfico, consultar o valor de cada ponto, entre outros.

Esta á a biblioteca utilizada para geração das visualizações gráficas de dados no Dash, um *framework* em Python para criação de *dashboards* e relatórios interativos *web* (*Javascript*).

* 1. CONSIDERAÇÃO FINAL

Para os fins deste trabalho, a biblioteca escolhida foi a Seaborn. Com a sua utilização, será possível gerar as visualizações escolhidas pelo usuário, contemplando todas aquelas descritas na Seção 4. Devido à fácil customização de seus gráficos e à habilidade de transformação das visualizações para arquivos de imagem (como o formato *.png*), suas saídas poderão ser utilizadas numa interface de usuário, seja esta construída em Python ou com *frameworks web* (e.g., *React*).

1. CONSTRUÇÃO DA FERRAMENTA

A ferramenta para auxílio e análise exploratória de *datasets* foi construída utilizando-se a linguagem Python e a biblioteca de visualização Plotly. As configurações do ambiente utilizado para desenvolvimento são as que seguem:

* Sistema Operacional: MAC OS Mojave – versão 10.14.2
* Processador e RAM: Intel Core i5, 8GB RAM
* Linguagem de Programação: Python – versão 3.6.5
* Bibliotecas utilizadas: Plotly – versão 3.8.0, Pandas – versão 0.23.0

Não foi desenvolvida uma interface gráfica para a ferramenta, entretanto, a mesma foi pensada de modo a permitir que seja acoplada a uma interface gráfica futuramente, seja esta construída em Python ou utilizando *frameworks web*, como Angular, React e outros.

Deste modo, após a escolha da linguagem Python, do levantamento das visualizações gráficas que seriam implementadas e da escolha da biblioteca Plotly, foram construídas as funções de impressão dos gráficos no *script* Python XXXX.py. Com isso, buscou-se garantir que as funções responsáveis pelas visualizações e importações de dados permanecessem isoladas, garantindo a possibilidade de acoplamento com uma interface gráfica.

Assim, um outro arquivo, XXX.py, foi implementado como a interface não gráfica com o usuário, para que fosse possível realizar os testes da ferramenta. Este *script* simplesmente implementa um *loop* para receber *inputs* do usuário através de teclado na linha de comando. Então, ele interpreta esses comandos, mais simples que as funções em Python, e cria os gráficos e visualizações solicitados pelo usuário, exportando-os em formato *html*.

1. RESULTADOS

XXXX

1. CONCLUSÕES

XXXX

1. BIBLIOGRAFIA

*Anderson, D.; Sweeney, D.; Williams, T.; Essentials Of Modern Business Statistics With Microsoft Office Excel – South-Western, Cengage Learning, 2011*

BERNARD MARR. Forbes Magazine. **How Much Data Do We Create Every Day?:**The Mind-Blowing Stats Everyone Should Read. Disponível em: <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#68fdc6a460ba>. Acesso em: 28 maio 2019.

FARRELL, Damien. DataExplore: An Application for General Data Analysis in Research and Education. **Journal Of Open Research Software**, [s.l.], v. 4, p.1-8, 22 mar. 2016. Ubiquity Press, Ltd.. <http://dx.doi.org/10.5334/jors.94>.

KDNUGGETS. **2019 KDnuggets Poll:**What software you used for Analytics, Data Mining, Data Science, Machine Learning projects in the past 12 months?. Disponível em: <https://www.kdnuggets.com/2019/05/new-poll-software-analytics-data-science-machine-learning.html>. Acesso em: 28 maio 2019.

*Provost, F.;Fawcett, T.; Data Science for Business – O’Reilly, 2013*

PYTHON. **Python's Homepage.**Disponível em: <https://www.python.org>. Acesso em: 9 jun. 2019.

*Seltman HJ (2012) Experimental design and analysis. Online* [*http://www.stat.cmu.edu/\*hseltman/309/Book/Book.pdf*](http://www.stat.cmu.edu/*hseltman/309/Book/Book.pdf)

TIOBE. **TIOBE index for May 2019.**Disponível em: <https://www.tiobe.com/tiobe-index/>. Acesso em: 01 jun. 2019.

*Tukey J (1977) Exploratory data analysis. Pearson, London*

*Freund J. E.; Simon, G.A. (2000) Estatística Aplicada – Economia, Administração e Contabilidade. 9a Edição, Bookman, 404 pg, ISBN 85-7307-531-7.*

*RIBECCA, Severino.****The Data Visualisation Catalogue.****Disponível em: <https://datavizcatalogue.com>. Acesso em: 15 jun. 2019.*

*SOUKUP, Tom; DAVIDSON, Ian.****Visual Data Mining:****Techniques and Tools for Data Visualization and Mining. Estados Unidos: Wiley Publishing, Inc., 2002.*

*HUNTER, John et al.****Matplotlib Homepage.****Disponível em: <https://matplotlib.org/index.html>. Acesso em: 18 maio 2019.*

WASKOM, Michael. **Seaborn Homepage.**Disponível em: <https://seaborn.pydata.org/#>. Acesso em: 18 maio 2019.

1. Homepage da ferramenta disponível em <https://powerbi.microsoft.com/pt-br/> [↑](#footnote-ref-1)
2. Homepage da ferramenta disponível em <https://www.tableau.com/pt-br> [↑](#footnote-ref-2)
3. *Dataset* disponível para *download* em: <https://www.kaggle.com/justinas/nba-players-data> [↑](#footnote-ref-3)
4. Documentação da biblioteca disponível em <https://github.com/mkleehammer/pyodbc/wiki> [↑](#footnote-ref-4)
5. Documentação da biblioteca disponível em <https://pandas.pydata.org/pandas-docs/stable/> [↑](#footnote-ref-5)