



INTENSIVÃO DE PYTHON {+}

100% ONLINE & GRATUITO

Apostila Completa Aula 2

Aprenda como fazer uma análise de dados
que vai deixar seu chefe impressionado!
Impressionador do absoluto zero!



Parte 1

Introdução

Introdução

O que vamos aprender

Na segunda aula do Intensivão do Python você vai aprender a criar um código de análise de dados. No dia a dia das empresas, é muito comum dúvidas sobre os resultados da empresa. Um conceito que cada dia mais cresce nas empresas é o **data driven**. Basicamente, é dizer que ações são tomadas com base nos dados e não em achismos. Aprenda como **fazer uma super análise do zero** com os conceitos abaixo:

Importando dados
de bases .csv

Tratar dados usando
a biblioteca Pandas

Importação de
bibliotecas

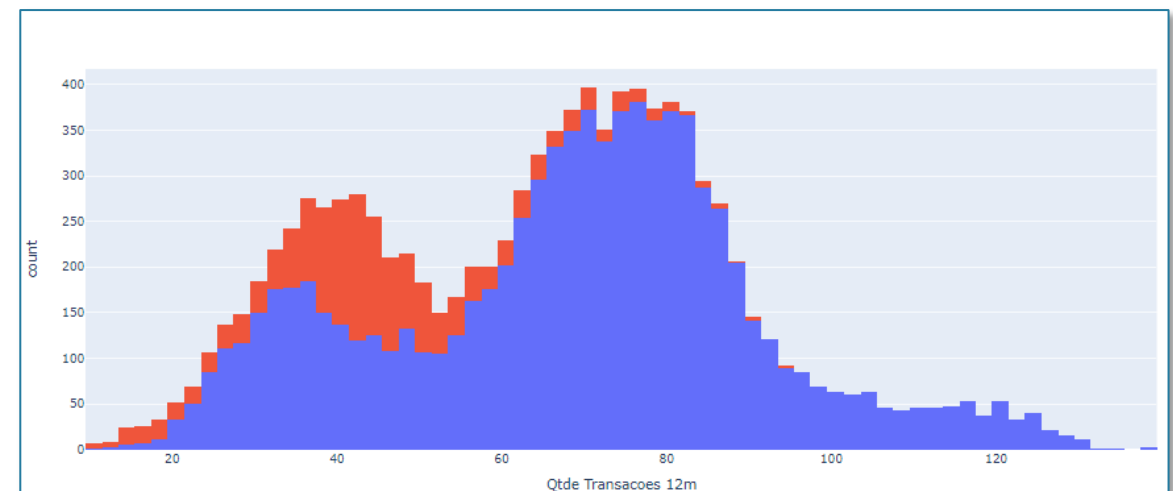
Criação de gráficos
usando o plotly

Após todos esses conhecimentos, seremos capazes de transformar uma tabela cheia de informações, nem um pouco fáceis de serem interpretadas ...

... em uma análise super aprofundada que servirão de base para tomada de decisão da gerência. Tudo graças a você! 😊

Prepare-se para **ver além do óbvio**.

```
ClientesBanco.csv - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
CLIENTNUM,Categoria,Idade,Sexo,Dependentes,Educação,Estado Civil,Faixa Salarial Anual,Categoria Cartão,Meses como Cliente,P
768805383,Cliente,45,M,3,Ensino Médio,Casado,$60K - $80K,Blue,39,5,1,3,12691,777,11914,1335,1144,42,1625,0.061
818770008,Cliente,49,F,5,Ensino Superior,Solteiro,Less than $40K,Blue,44,6,1,2,8256,864,7392,1541,1291,33,3714,0.105
713982108,Cliente,51,M,3,Ensino Superior,Casado,$80K - $120K,Blue,36,4,1,0,3418,0,3418,2594,1887,20,2333,0
769911858,Cliente,40,F,4,Ensino Médio,Não informado,Less than $40K,Blue,34,3,4,1,3313,2517,796,1405,1171,20,2333,0.76
709106358,Cliente,40,M,3,Sem ensino formal,Casado,$60K - $80K,Blue,21,5,1,0,4716,0,4716,2175,816,28,2,5,0
713061558,Cliente,44,M,2,Ensino Superior,Casado,$40K - $60K,Blue,36,3,1,2,4010,1247,2763,1376,1088,24,0,846,0.311
819347208,Cliente,51,M,4,Não informado,Casado,$120K +,Gold,46,6,1,3,34516,2264,32252,1975,1330,31,0,722,0.066
818906208,Cliente,32,M,0,Ensino Médio,Não informado,$60K - $80K,Silver,27,2,2,29081,1396,27685,2204,1538,36,0,714,0.048
710930508,Cliente,37,M,3,Sem ensino formal,Solteiro,$60K - $80K,Blue,36,5,2,0,22352,2517,19835,3355,1350,24,1182,0.113
719661558,Cliente,48,M,2,Ensino Superior,Solteiro,$80K - $120K,Blue,36,6,3,3,11656,1677,9979,1524,1441,32,0,882,0.144
708790833,Cliente,42,M,5,Sem ensino formal,Não informado,$120K +,Blue,31,5,3,2,6748,1467,5281,0,831,1201,42,0,68,0.217
710821833,Cliente,65,M,1,Não informado,Casado,$40K - $60K,Blue,54,6,2,3,9095,1587,7508,1433,1314,26,1364,0.174
710599683,Cliente,56,M,1,Ensino Superior Incompleto,Solteiro,$80K - $120K,Blue,36,3,6,0,11751,0,11751,3397,1539,17,3,25,0
816082233,Cliente,35,M,3,Ensino Superior,Não informado,$60K - $80K,,30,5,1,3,8547,1666,6881,1163,1311,33,2,0,195
712396908,Cliente,57,F,2,Ensino Superior,Casado,Less than $40K,Blue,48,5,2,2,2436,680,1756,1,19,1570,29,0,611,0.279
714885258,Cliente,44,M,4,Não informado,Não informado,$80K - $120K,Blue,37,5,1,2,4234,972,3262,1707,1348,27,1,7,0.23
709967358,Cliente,48,M,4,Post-Ensino Superior,Solteiro,$80K - $120K,Blue,36,6,2,3,30367,2362,28005,1708,1671,27,0,929,0.078
753327333,Cliente,41,M,3,Não informado,Casado,$80K - $120K,Blue,34,4,4,1,13535,1291,12244,0,653,1028,21,1625,0.095
806160108,Cliente,61,M,1,Ensino Médio,Casado,$40K - $60K,Blue,56,2,2,3,3193,2517,676,1831,1336,30,1143,0.788
709327383,Cliente,45,F,2,Ensino Superior,Casado,Não informado,Blue,37,6,1,2,14470,1157,13313,0,966,1207,21,0,909,0.08
806165208,Cliente,47,M,1,Doutorado,Divorciado,$60K - $80K,Blue,42,5,2,0,20979,1800,19179,0,906,1178,27,0,929,0.086
708508758,Cancelado,62,F,0,Ensino Superior,Casado,Less than $40K,Blue,49,2,3,3,1438,3,0,1438,3,1047,692,16,0,6,0
784725333,Cliente,41,M,3,Ensino Médio,Casado,$40K - $60K,Blue,33,4,2,1,4470,680,3790,1608,931,18,1571,0.152
811604133,Cliente,47,F,4,Não informado,Solteiro,Less than $40K,Blue,36,3,3,2,2492,1560,932,0,573,1126,23,0,353,0.626
789124683,Cliente,54,M,2,Não informado,Casado,$80K - $120K,Blue,42,4,2,3,12217,0,12217,1075,1110,21,0,75,0
771071958,Cliente,41,F,3,Ensino Superior,Solteiro,Less than $40K,Blue,28,6,1,2,7768,1669,6099,0,797,1051,22,0,833,0.215
720466383,Cliente,59,M,1,Ensino Médio,Não informado,$40K - $60K,Blue,46,4,1,2,14784,1374,13410,0,921,1197,23,1,3,0.093
804424383,Cliente,63,M,1,Não informado,Casado,$60K - $80K,Blue,56,3,3,2,10215,1010,9205,0,843,1904,40,1,0,099
718813833,Cliente,44,F,3,Sem ensino formal,Solteiro,Não informado,Blue,34,5,2,2,10100,0,10100,0,525,1052,18,1571,0
806624208,Cliente,47,M,4,Ensino Médio,Casado,$40K - $60K,Blue,42,6,0,0,4785,1362,3423,0,739,1045,38,0,9,0.285
778348233,Cliente,53,M,3,Não informado,Casado,$80K - $120K,Blue,33,3,2,3,2753,1811,942,0,977,1038,25,2571,0.658
```



Entendendo a base de dados

As informações que vão alimentar nossa análise, foram extraídas do site Kaggle([link](#)). Os dados são referentes a clientes de cartão de crédito e seus hábitos de consumo, reclamações, etc.

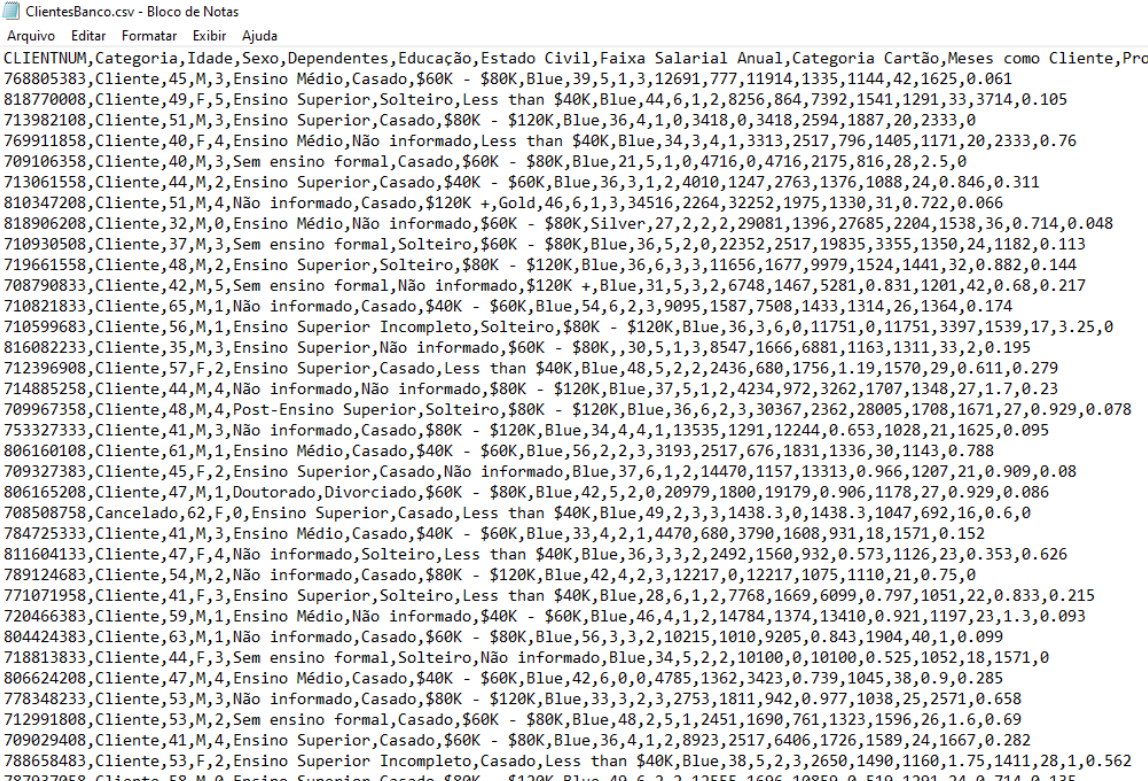
A imagem ao lado, mostra os dados extraídos em modelo **.csv**. Como podemos ver, os dados não estão formatados o que nos dificulta um pouco entender corretamente o que temos aqui...

A situação é a seguinte:

Seu chefe, após olhar os resultados do trimestre, não consegue entender os motivos para os clientes do cartão de crédito estarem cancelando seus cartões.

Sem te dar nenhum direcionamento, ele pede que você faça uma análise que possa ajudá-lo a entender/resolver o problema.

A única informação que você tem é um arquivo **.csv** extraído do sistema da empresa (apresentado ao lado).



ClientesBanco.csv - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

CLIENTNUM,Categoria,Idade,Sexo,Dependentes,Educação,Estado Civil,Faixa Salarial Anual,Categoria Cartão,Meses como Cliente,Pro

768805383,Cliente,45,M,3,Ensino Médio,Casado,\$60K - \$80K,Blue,39,5,1,3,12691,777,11914,1335,1144,42,1625,0.061

818770008,Cliente,49,F,5,Ensino Superior,Solteiro,Less than \$40K,Blue,44,6,1,2,8256,864,7392,1541,1291,33,3714,0.105

713982108,Cliente,51,M,3,Ensino Superior,Casado,\$80K - \$120K,Blue,36,4,1,0,3418,0,3418,2594,1887,20,2333,0

769911858,Cliente,40,F,4,Ensino Médio,Não informado,Less than \$40K,Blue,34,3,4,1,3313,2517,796,1405,1171,20,2333,0.76

709106358,Cliente,40,M,3,Sem ensino formal,Casado,\$60K - \$80K,Blue,21,5,1,0,4716,0,4716,2175,816,28,2.5,0

713061558,Cliente,44,M,2,Ensino Superior,Casado,\$40K - \$60K,Blue,36,3,1,2,4010,1247,2763,1376,1088,24,0.846,0.311

810347208,Cliente,51,M,4,Não informado,Casado,\$120K +,Gold,46,6,1,3,34516,2264,32252,1975,1330,31,0.722,0.066

818906208,Cliente,32,M,0,Ensino Médio,Não informado,\$60K - \$80K,Silver,27,2,2,2,29081,1396,27685,2204,1538,36,0.714,0.048

710930508,Cliente,37,M,3,Sem ensino formal,Solteiro,\$60K - \$80K,Blue,36,5,2,0,22352,2517,19835,3355,1350,24,1182,0.113

719661558,Cliente,48,M,2,Ensino Superior,Solteiro,\$80K - \$120K,Blue,36,6,3,3,11656,1677,9979,1524,1441,32,0.882,0.144

708790833,Cliente,42,M,5,Sem ensino formal,Não informado,\$120K +,Blue,31,5,3,2,6748,1467,5281,0.831,1201,42,0.68,0.217

710821833,Cliente,65,M,1,Não informado,Casado,\$40K - \$60K,Blue,54,6,2,3,9095,1587,7508,1433,1314,26,1364,0.174

710599683,Cliente,56,M,1,Ensino Superior Incompleto,Solteiro,\$80K - \$120K,Blue,36,3,6,0,11751,0,11751,3397,1539,17,3.25,0

816082233,Cliente,35,M,3,Ensino Superior,Não informado,\$60K - \$80K,,30,5,1,3,8547,1666,6881,1163,1311,33,2,0.195

712396908,Cliente,57,F,2,Ensino Superior,Casado,Less than \$40K,Blue,48,5,2,2,2436,680,1756,1.19,1570,29,0.611,0.279

714885258,Cliente,44,M,4,Não informado,Não informado,\$80K - \$120K,Blue,37,5,1,2,4234,972,3262,1707,1348,27,1.7,0.23

709967358,Cliente,48,M,4,Post-Ensino Superior,Solteiro,\$80K - \$120K,Blue,36,6,2,3,30367,2362,28005,1708,1671,27,0.929,0.078

753327333,Cliente,41,M,3,Não informado,Casado,\$80K - \$120K,Blue,34,4,4,1,13535,1291,12244,0.653,1028,21,1625,0.095

806160108,Cliente,61,M,1,Ensino Médio,Casado,\$40K - \$60K,Blue,56,2,2,3,3193,2517,676,1831,1336,30,1143,0.788

709327383,Cliente,45,F,2,Ensino Superior,Casado,Não informado,Blue,37,6,1,2,14470,1157,13313,0.966,1207,21,0.909,0.08

806165208,Cliente,47,M,1,Doutorado,Divorciado,\$60K - \$80K,Blue,42,5,2,0,20979,1800,19179,0.906,1178,27,0.929,0.086

708508758,Cancelado,62,F,0,Ensino Superior,Casado,Less than \$40K,Blue,49,2,3,3,1438.3,0,1438.3,1047,692,16,0.6,0

784725333,Cliente,41,M,3,Ensino Médio,Casado,\$40K - \$60K,Blue,33,4,2,1,4470,680,3790,1608,931,18,1571,0.152

811604133,Cliente,47,F,4,Não informado,Solteiro,Less than \$40K,Blue,36,3,3,2,2492,1560,932,0.573,1126,23,0.353,0.626

789124683,Cliente,54,M,2,Não informado,Casado,\$80K - \$120K,Blue,42,4,2,3,12217,0,12217,1075,1110,21,0.75,0

771071958,Cliente,41,F,3,Ensino Superior,Solteiro,Less than \$40K,Blue,28,6,1,2,7768,1669,6099,0.797,1051,22,0.833,0.215

720466383,Cliente,59,M,1,Ensino Médio,Não informado,\$40K - \$60K,Blue,46,4,1,2,14784,1374,13410,0.921,1197,23,1.3,0.093

804424383,Cliente,63,M,1,Não informado,Casado,\$60K - \$80K,Blue,56,3,3,2,10215,1010,9205,0.843,1904,40,1,0.099

718813833,Cliente,44,F,3,Sem ensino formal,Solteiro,Não informado,Blue,34,5,2,2,10100,0,10100,0.525,1052,18,1571,0

806624208,Cliente,47,M,4,Ensino Médio,Casado,\$40K - \$60K,Blue,42,6,0,0,4785,1362,3423,0.739,1045,38,0.9,0.285

778348233,Cliente,53,M,3,Não informado,Casado,\$80K - \$120K,Blue,33,3,2,3,2753,1811,942,0.977,1038,25,2571,0.658

712991808,Cliente,53,M,2,Sem ensino formal,Casado,\$60K - \$80K,Blue,48,2,5,1,2451,1690,761,1323,1596,26,1.6,0.69

709029408,Cliente,41,M,4,Ensino Superior,Casado,\$60K - \$80K,Blue,36,4,1,2,8923,2517,6406,1726,1589,24,1667,0.282

788658483,Cliente,53,F,2,Ensino Superior Incompleto,Casado,Less than \$40K,Blue,38,5,2,3,2650,1490,1160,1.75,1411,28,1,0.562

787937058,Cliente,58,M,0,Ensino Superior,Casado,\$80K - \$120K,Blue,49,6,2,2,12555,1696,10859,0.519,1291,24,0.714,0.135

Entendendo a solução final

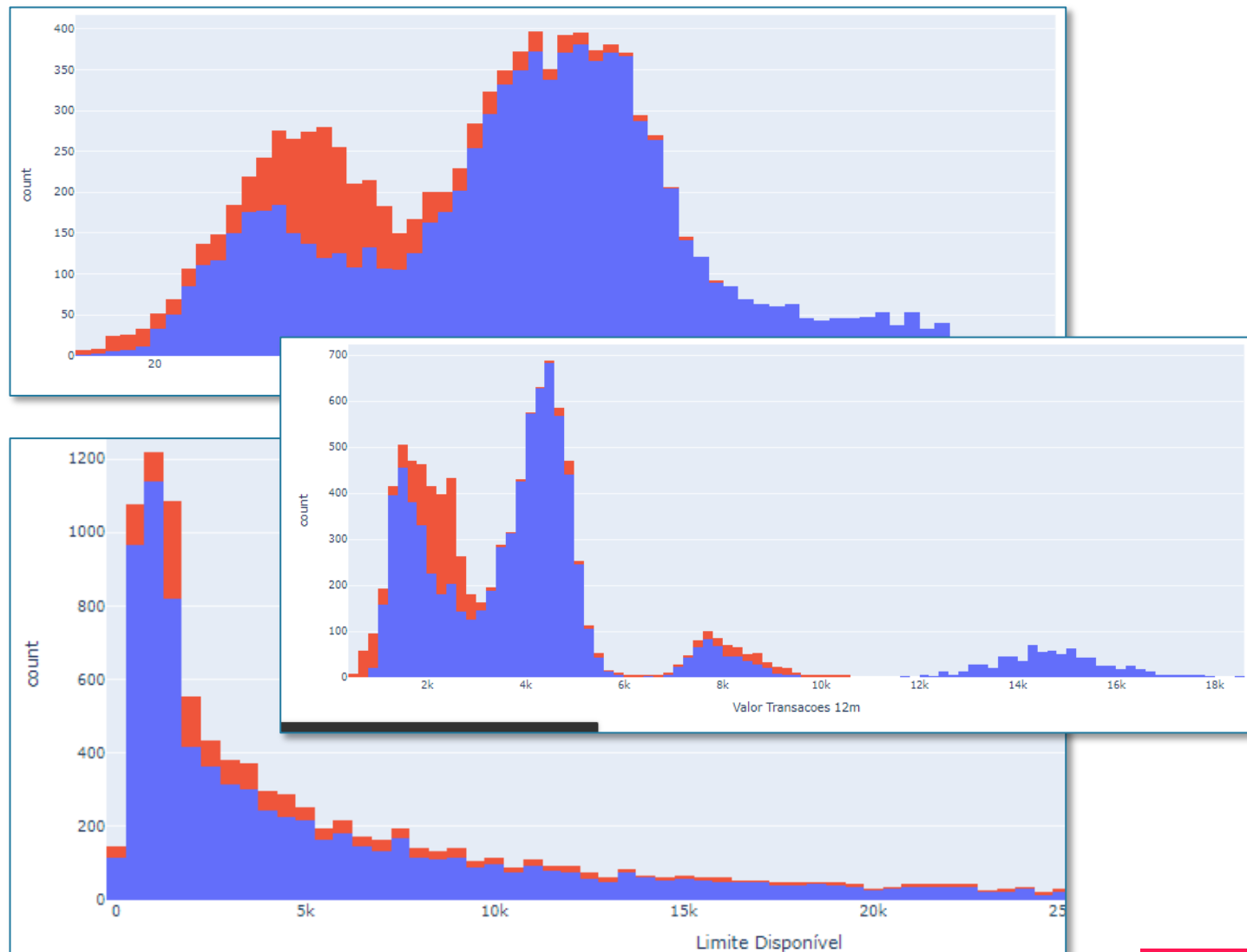
Nesse caso, a solução final podem ser diversas!

Estamos aqui tratando de análise de dados. Boa parte da solução aqui é não conhecida nesse estágio.

Quando encontramos esses casos é muito importante, mais até do que a solução em um primeiro momento, qual o problema.

Na aula e na apostila vamos fazer uma análise exploratória de dados e buscar direcionadores de causas raiz que podem ser atacadas visando o maior retorno com o menor esforço.

Para isso utilizaremos o Python para nos ajudar em análises gráficas dos dados como este aqui do lado 😊



Parte 2

Importando e visualizando os dados

Importando base de dados (1/2)

Como vimos na aula 1 do Intensivão, vamos usar bibliotecas que nos facilitem importar dados de planilhas Excel, arquivos .csv, etc.

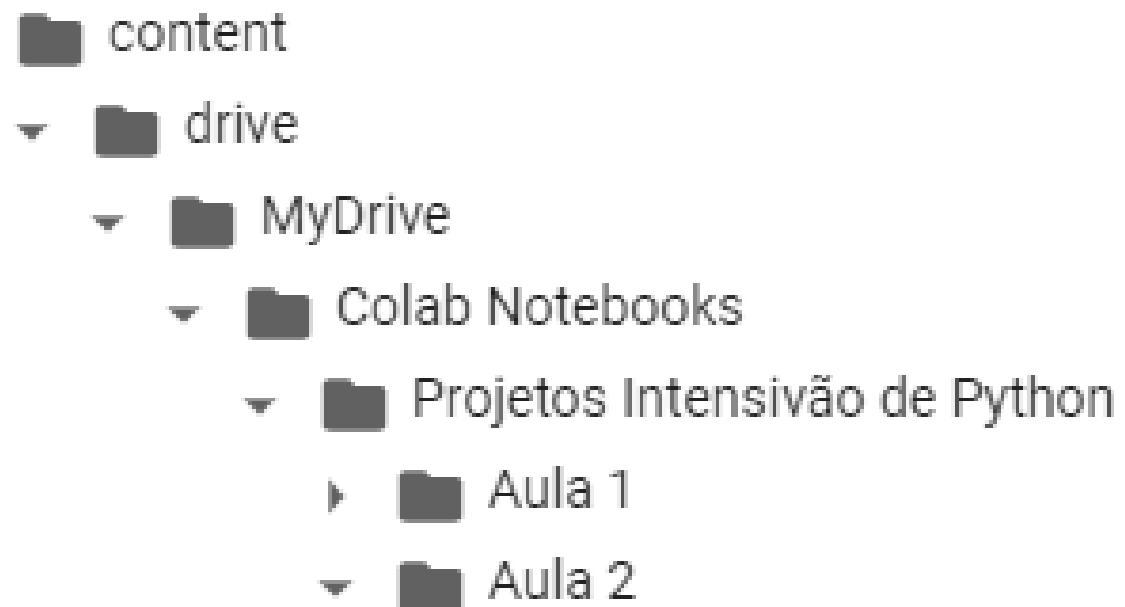
Novamente usaremos o PANDAS. Caso você não saiba do que estamos falando aqui, dá uma olhadinha na apostila da Aula 1 do intensivão!! Lá a gente explica o que são bibliotecas e para que servem 😊.

Vamos começar importando o PANDAS como pd.

Feito isso, precisamos agora buscar o arquivo no nosso pc.

Aqui no intensivão vamos usar o caminho ao lado:

```
import pandas as pd
```



`/content/drive/MyDrive/Colab Notebooks/Projetos Intensivão de Python/Aula 2/ClientesBanco.csv'`

Importando base de dados (2/2)

Se você acompanhou a aula 1 do Intensivão, vai lembrar que lá usamos a função `read_excel()` do Pandas. Aqui, temos uma diferença. Como se trata de um arquivo `.csv`, precisamos usar a **fórmula `.read_csv()`** conforme apresentado abaixo:

```
import pandas as pd
clientes_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Projetos Intensivão de Python/Aula 2/ClientesBanco.csv', encoding='latin1')
```

Lê o arquivo `.csv` indicado dentro do parênteses

Argumento necessário para bases com caracteres como `'ç', '~'`.

Local onde o arquivo `.csv` se encontra

Nome do arquivo `.csv`

Até aqui, nada muito diferente... Apenas se atente a última parte do parênteses : **`encoding = 'latin1'`**.

Como o Python se utiliza de caracteres do inglês, ele não possui por *default* caracteres como `'ç', '~', '^'`, que para nós brasileiros é normal.

Por isso, quando lidamos com bases que possam ter estes tipos de caracteres, é importante usarmos este argumento ao fim do `read_csv`.

Priorizando os dados importados (1/2)

Cada vez mais, saber que dados são úteis ou não é algo fundamental no dia a dia do trabalho.

É muito comum, termos bases de dados ENORMES extraídas de sistemas.

Saber **separar o que é útil do que não é**, é fundamental para uma boa análise de dados.

Vamos voltar para nosso exemplo, sabemos que o chefe não está perguntando o motivo do FULANO ter cancelado o cartão e sim porque temos CLIENTES cancelando o cartão....

Portanto, ao darmos uma olhada rápida na nossa base(print ao lado) podemos ver que temos uma coluna **CLIENTENUM**. Não nos é interessante pois é uma informação irrelevante para nosso estudo.

Sendo assim, podemos retirá-la da nossa tabela. Vamos ver como, a seguir.

	CLIENTNUM	Categoria	Idade	Sexo	De
0	768805383	Cliente	45	M	
1	818770008	Cliente	49	F	
2	713982108	Cliente	51	M	
3	769911858	Cliente	40	F	
4	709106358	Cliente	40	M	
...
10122	772366833	Cliente	50	M	
10123	710638233	Cancelado	41	M	

A coluna CLIENTNUM não nos é relevante.

Podemos retirá-la de nossa base para aumentar a eficiência do código.

Priorizando os dados importados (2/2)

Para retirarmos a coluna **CLIENTNUM**, vamos usar o método abaixo:

.drop()

Este método será aplicado na variável **clientes_df** criada na primeira linha do código para receber os dados do arquivo .csv.

Este método necessitará de alguns argumentos:

- Nome da coluna ou código da linha a ser removida: ('**CLIENTNUM**')
- Qual dos eixos deve ser excluído:
 - **0** ou '**index**' será apagada a linha indicada;
 - **1** ou '**columns**' será apagada a coluna indicada.

```
import pandas as pd

clientes_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Projetos')
clientes_df = clientes_df.drop('CLIENTNUM', axis=1)
```

	CLIENTNUM	Categoria	Idade
0	768805383	Cliente	45
1	81171308	Cliente	49
2	713982108	Cliente	51

	Categoria	Idade	Sexo
0	Cliente	45	M
1	Cliente	49	F
2	Cliente	51	M

Coluna CLIENTNUM retirada

Visualizar a base de dados importada

Bem, já importamos nossa base de dados...

Agora vamos tentar visualizá-la !

Usamos a função **DISPLAY()** para exibir nossos dados coletados.

Lembra como era difícil entender os dados quando estavam em .csv?

Com o pandas essa visualização fica bem mais amigável e prática.

Perceba também que já excluímos a coluna CLIENTNUM que não queríamos.

IMPORTANTE: A base original, **NÃO** foi afetada.

```
clientes_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Projetos Intensivo de Python/Aula 2/ClientesBanco.csv',
clientes_df = clientes_df.drop('CLIENTNUM', axis=1)
display(clientes_df)
```

Função display que apresenta os dados armazenados na variável df

	Categoria	Idade	Sexo	Dependentes	Educação	Estado Civil	Faixa Salarial Anual	Categoria Cartão	Meses como Cliente	Produtos Contratados	Inatividade 12m	Con
0	Cliente	45	M	3	Ensino Médio	Casado	\$60K - \$80K	Blue	39	5	1	
1	Cliente	49	F	5	Ensino Superior	Solteiro	Less than \$40K	Blue	44	6	1	
2	Cliente	51	M	3	Ensino Superior	Casado	\$80K - \$120K	Blue	36	4	1	
3	Cliente	40	F	4	Ensino Médio	Não informado	Less than \$40K	Blue	34	3	4	
4	Cliente	40	M	3	Sem ensino formal	Casado	\$60K - \$80K	Blue	21	5	1	
...
10122	Cliente	50	M	2	Ensino Superior	Solteiro	\$40K - \$60K	Blue	40	3	2	
10123	Cancelado	41	M	2	Não informado	Divorciado	\$40K - \$60K	Blue	25	4	2	
10124	Cancelado	44	F	1	Ensino Médio	Casado	Less than \$40K	Blue	36	5	3	
10125	Cancelado	30	M	2	Ensino Superior	Não informado	\$40K - \$60K	Blue	36	4	3	
10126	Cancelado	43	F	2	Ensino Superior	Casado	Less than \$40K	Silver	25	6	2	

10127 rows x 20 columns

10127 linhas
20 colunas

Parte 3

Tratamento e visão geral dos dados

Tratamento e visão geral dos dados

Limpando a base de dados

É muito comum que base de dados extraídas de sistemas possuam dados faltantes e/ou dados que não são corretos.

Todos esses dados influenciam diretamente nos resultados obtidos na nossa análise.

Imagine um **caso genérico** em que preciso calcular a média de consumo de cartões de crédito. Caso existam dados faltantes que possam ser considerados como valor 0, meu cálculo de média será afetado diretamente.

Assim, é sempre importante antes de qualquer análise avaliar se precisamos tratar esta base de dados ou não.

Para o caso descrito acima, que temos dados faltantes, usaremos uma variação do método drop:

.dropna()

Este método retirará linhas que possuam dados vazios/faltantes..

```
clientes_df = clientes_df.dropna()
display(clientes_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10126 entries, 0 to 10126
Data columns (total 20 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Categoria                               10126 non-null  object
1   Idade                                   10126 non-null  int64
2   Sexo                                   10126 non-null  object
3   Dependentes                             10126 non-null  int64
4   Educação                               10126 non-null  object
5   Estado Civil                             10126 non-null  object
6   Faixa Salarial Anual                     10126 non-null  object
7   Categoria Cartão                         10126 non-null  object
8   Meses como Cliente                       10126 non-null  int64
9   Produtos Contratados                     10126 non-null  int64
10  Inatividade 12m                          10126 non-null  int64
11  Contatos 12m                             10126 non-null  int64
12  Limite                                    10126 non-null  float64
13  Limite Consumido                         10126 non-null  int64
14  Limite Disponível                        10126 non-null  float64
15  Mudanças Transacoes_Q4_Q1               10126 non-null  float64
16  Valor Transacoes 12m                     10126 non-null  int64
17  Qtde Transacoes 12m                     10126 non-null  int64
18  Mudança Qtde Transações_Q4_Q1           10126 non-null  float64
19  Taxa de Utilização Cartão                10126 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.6+ MB
None
```

Número de
linhas

10126
linhas são
não Nulas.

Análise descritiva dos dados

Quando estamos trabalhando com **grande quantidade de dados** que não conhecemos a fundo, é interessante fazer algumas análises exploratórias que nos permitam entender um pouco melhor como estão distribuídos esses dados.

Uma das formas de fazer essa análise exploratória, é por meio da estatística.

No Pandas vamos usar o método abaixo:

.describe()

Esse método, nos fornece as informações abaixo de cada uma das colunas existentes:

- **Count:** Número de registros na linha;
- **Mean :** Média dos valores;
- **Std:** Desvio Padrão dos valores;
- **Min:** Menor valor entre os dados;
- **25%:** 1ºQuartil - Valor onde temos 25% dos dados;
- **50%:** 2ºQuartil - Valor onde temos 50% dos dados;
- **75%:** 3ºQuartil - Valor onde temos 75% dos dados;
- **Max:** Maior valor entre os dados

```
clientes_df = clientes_df.dropna()
display(clientes_df.info())
display(clientes_df.describe())
```

	Idade	Dependentes	Meses como Cliente	Produtos Contratados	Inatividade 12m
count	10126.000000	10126.000000	10126.000000	10126.000000	10126.000000
mean	46.327079	2.346139	35.928995	3.812463	2.341300
std	8.016420	1.298956	7.986593	1.554440	1.010584
min	26.000000	0.000000	13.000000	1.000000	0.000000
25%	41.000000	1.000000	31.000000	3.000000	2.000000
50%	46.000000	2.000000	36.000000	4.000000	2.000000
75%	52.000000	3.000000	40.000000	5.000000	3.000000
max	73.000000	5.000000	56.000000	6.000000	6.000000

O .describe nos fornece uma análise descritiva dos dados e de sua distribuição.

Divisão de clientes e cancelados

Na coluna 'Categoria', temos 2 opções de resultados:

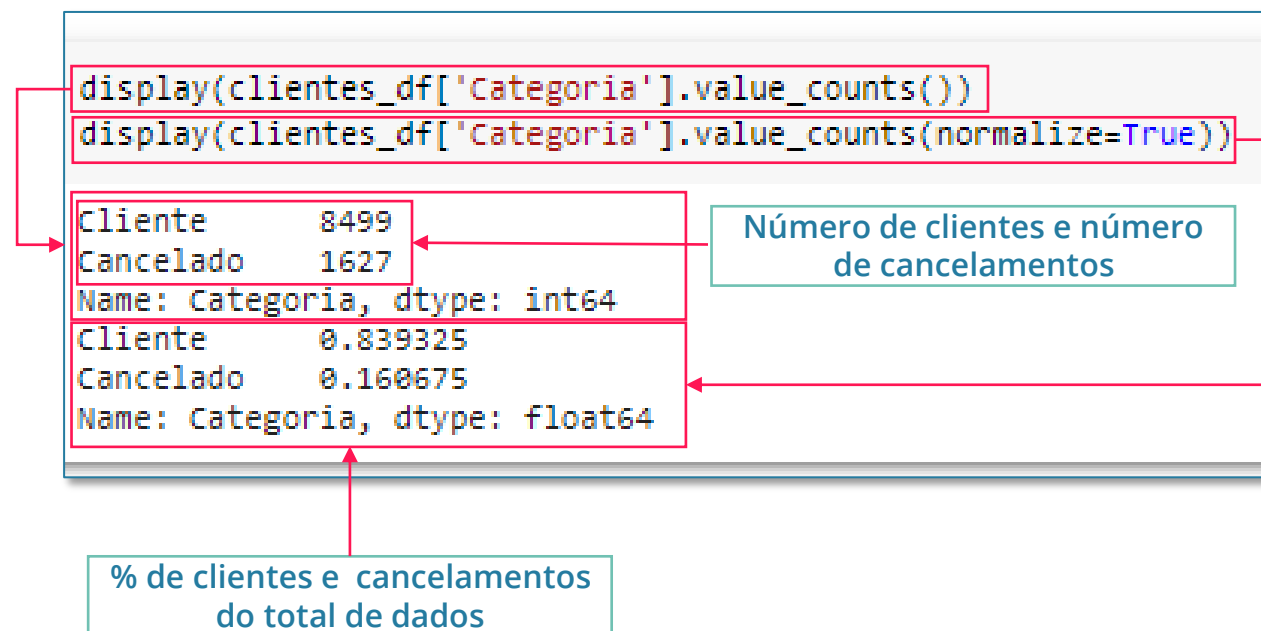
- Cliente;
- Cancelado.

Como é de nosso interesse entender a motivação dos cancelamentos, é interessante dividirmos estes dois grupos.

Para isso, vamos usar mais um método do pandas na nossa base clientes_df:

`.value_counts()`

Perceba que usamos o termo **['Categoria']** para indicar que estamos usando apenas esta coluna da base de dados armazenada na variável clientes_df.



Parte 4

Analizando os dados

Analizando os dados

Como seguir?

O que vamos ver daqui para a frente, vai além do Python em si.

Poderíamos explicar milhões de formas de analisar os dados do Python, mas uma coisa é essencial:

O QUE EU QUERO RESPONDER?

Entender seu problema é fundamental. Assim, será possível orientar sua análise para resolver o problema.

Então vamos lá!

Nosso problema é:

“Aumento do cancelamento de cartão de crédito.”

O que eu quero:

“Entender os principais motivos que levam ao cancelamento para assim gerar um plano de ação”



Análise gráfica – criando uma função (1/2)

Um dos caminhos mais comuns e usuais para analisarmos os dados é através de uma análise gráfica.

Temos um total de 20 colunas, cada uma delas nos fornece uma informação distinta. Fazer análise gráfica destes dados pode ser um tanto quanto repetitiva por serem muitos dados.

Para fazermos isso um pouco mais rápido vamos criar uma função que gera gráficos automaticamente via Python.

Se você não sabe o que significa criar uma função, dá uma olhadinha na apostila da aula 1 do intensivão. Lá, explicamos um pouco mais 😊

O nome da nossa função será **gráfico_coluna_categoria** com 2 argumentos **coluna** e **tabela**.

Vamos importar mais uma biblioteca para nos ajudar neste processo: **plotly.express**

Importando biblioteca plotly.express

```
import plotly.express as px
```

Criação da função

```
def grafico_coluna_categoria(coluna, tabela):  
    fig = px.histogram(tabela, x=coluna, color='Categoria')  
    fig.show()
```

Indentação representando que essas linhas pertencem ao def

Análise gráfica – criando uma função (2/2)

Vamos continuar a avaliar as outras linhas de código da nossa função.

Primeiramente, vamos precisar criar uma variável **fig** que receberá os dados a serem printados.

Esses dados (gráfico) serão calculados a partir do uso do método **.histogram()**.

Podemos ver que os argumentos necessários são :

- **Tabela:** argumento da função, deverá ser fornecido no momento de ativação da função pelo usuário;
- **X=coluna:** Serão dados do eixo X. Os valores que existem na coluna fornecida pelo usuário no momento da ativação da função;
- **Color='Categoria':** O gráfico terá cores diferentes para diferentes valores da coluna 'Categoria'

```
import plotly.express as px
```

```
def grafico_coluna_categoria(coluna, tabela):
```

```
    fig = px.histogram(tabela, x=coluna, color='Categoria')
```

```
    fig.show()
```

Método .show para que seja exibido as informações da variável fig

Biblioteca plotly express

Método e seus argumentos

Variável para recebimento das informações calculadas

Criando os gráficos (1/4)

Agora que temos uma função que nos auxilia na criação de gráficos, podemos criar uma linha de código que gera os gráficos para cada uma das colunas existentes.

Para isso, vamos usar o conceito de estrutura de repetição por meio do FOR.

Categoria	Idade	Sexo	Dependentes	Educação	Estado Civil	Faixa Salarial Anual	Categoria Cartão
Cliente	45	M	3	Ensino Médio	Casado	\$60K - \$80K	Blue
Cliente	49	F	5	Ensino Superior	Solteiro	Less than \$40K	Blue
Cliente	51	M	3	Ensino Superior	Casado	\$80K - \$120K	Blue
Cliente	40	F	4	Ensino Médio	Não Informado	Less than \$40K	Blue

FOR coletará todos as colunas da tabela clientes_df

```
for coluna in clientes_df:
```

```
    grafico_coluna_categoria(coluna, clientes_df)
```

Indentação For

Usando a função gráfico_coluna_categoria para a base clientes_df

Criando os gráficos (2/4)

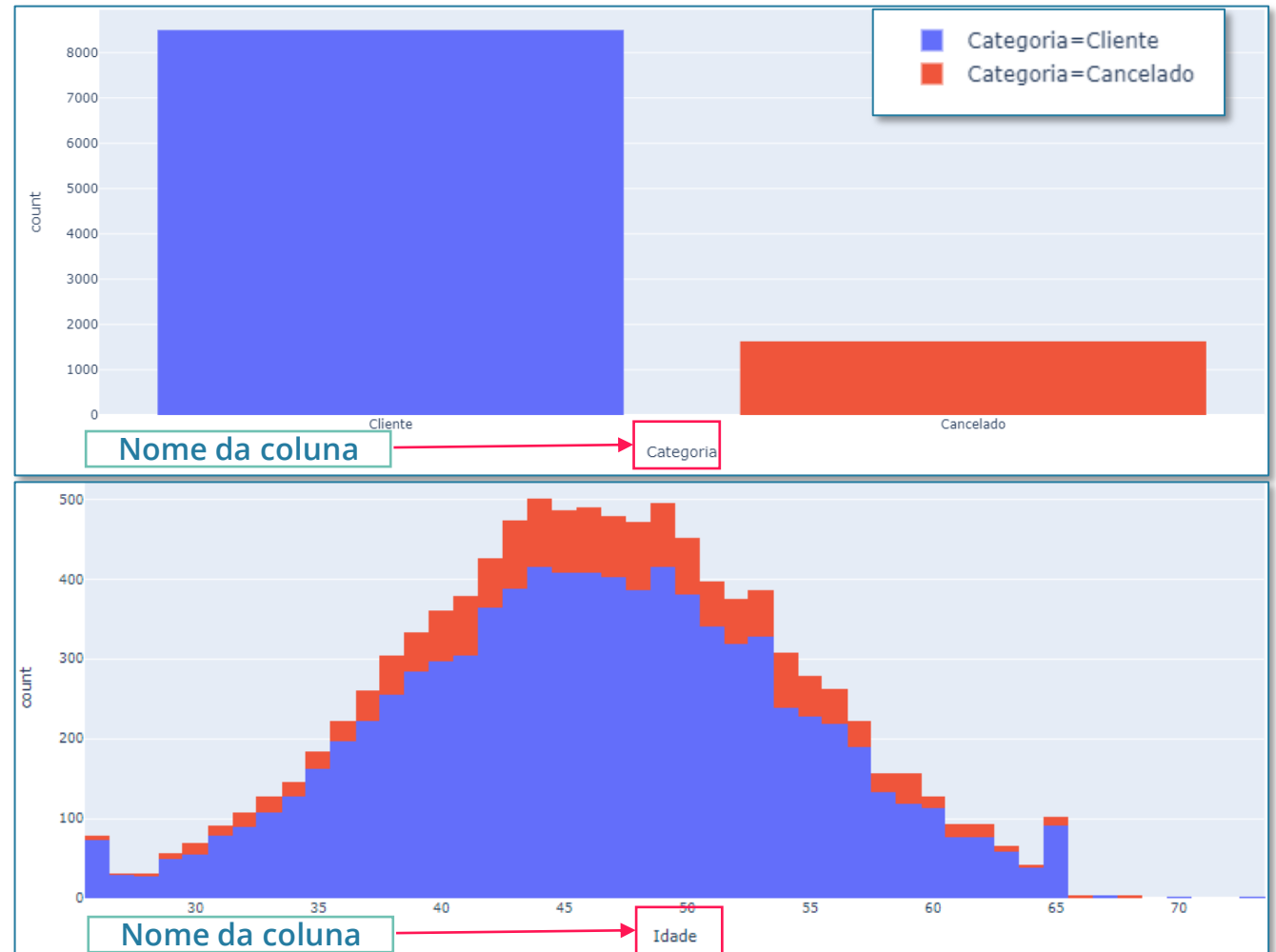
Como foi dito, em apenas um for conseguimos criar todos os gráficos disponíveis.

O eixo x sempre será o nome da coluna. Isso já era esperado visto que ao fazermos a nossa função parametrizamos **x=coluna**:

```
def grafico_coluna_categoria(coluna, tabela):
    fig = px.histogram(tabela, x=coluna, color='Categoria')
```

Além disso, podemos perceber que os dados que possuem valor da coluna Categoria como cancelado, são representados em uma cor distinta aos clientes. Isso também era esperado visto a parametrização **color='Categoria'**

```
def grafico_coluna_categoria(coluna, tabela):
    fig = px.histogram(tabela, x=coluna, color='Categoria')
```



Criando os gráficos (3/4)

Assim, temos vários outros gráficos que nos permitem analisar de maneira mais profunda o nosso problema. Boa parte do problema daqui para a frente é muito mais uma questão de análise do que Python propriamente dita.

Por exemplo, dos 4 gráficos abaixo podemos perceber que existe uma concentração de cancelamentos relacionados a Quantidade de transações. Essa informação pode ser o início de uma análise mais aprofundada.



Criando os gráficos (4/4)

Voltando para a criação de gráficos.

É possível que você queira customizar seus gráficos para que fiquem mais atrativos ou para exibir alguma informação do seu interesse.

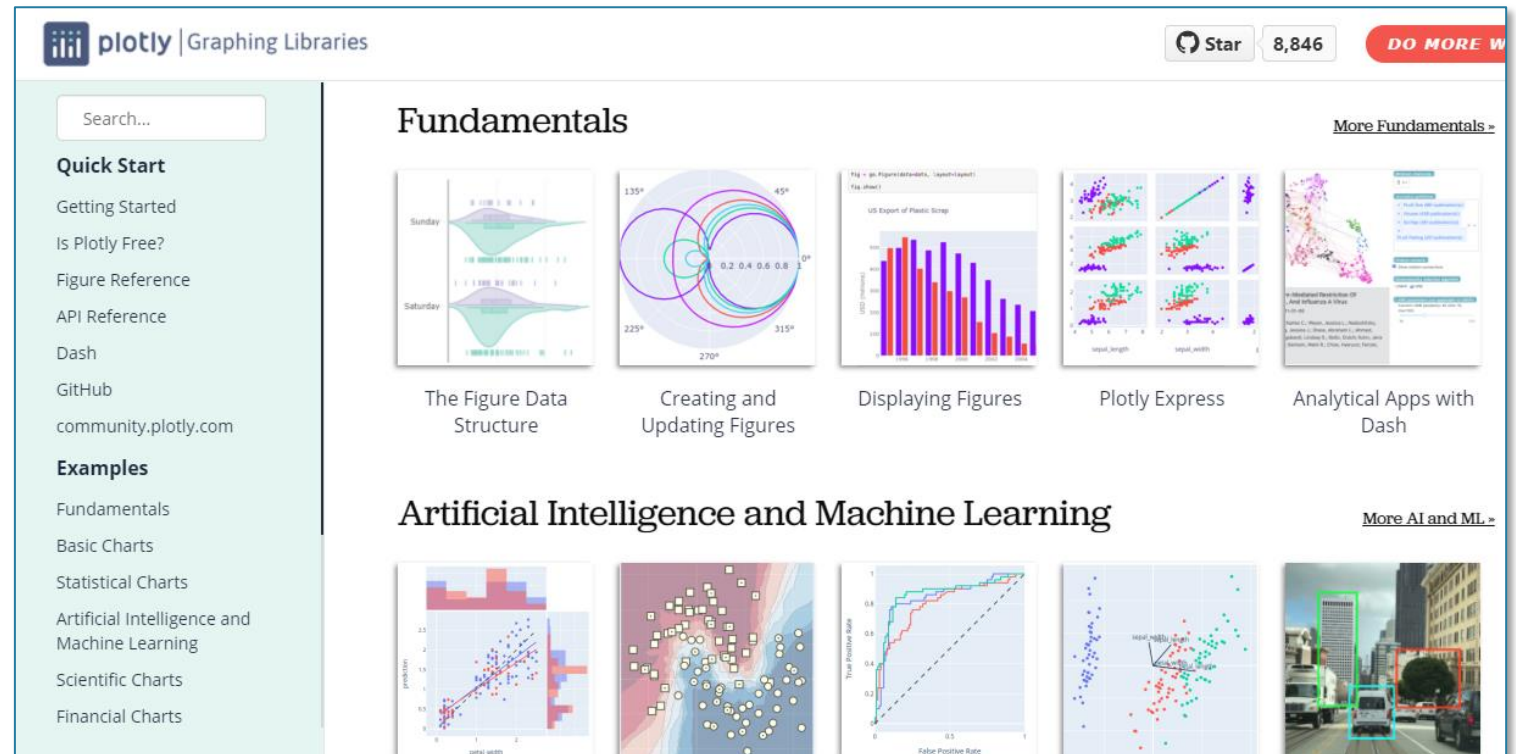
Vale lembrar que utilizamos uma biblioteca para a criação destes gráficos, o plotly:

O plotly por si só possui uma documentação vasta e detalhada de todas as customizações e alterações que a biblioteca permite.

Caso você tenha interesse é só acessar:

Para todos tipos de gráficos:
<https://plotly.com/python/>

Para nosso caso específico de histograma:
<https://plotly.com/python/histograms/>



INTENSIVÃO DE PYTHON {#}

100% ONLINE & GRATUITO

Ainda não segue a gente no Instagram e nem é inscrito no nosso canal do Youtube? Então corre lá!



@hashtagprogramacao



youtube.com/hashtag-programacao

