DESAFIO INDICIUM

ESTAGIO DATA ENGINEERING

Relatório de Atividade

Ricardo Castilhos Thisted

Florianópolis – Agosto de 2020

Para começar com o desafio, iniciei importando as bibliotecas que julguei necessárias para a realização de uma tarefa de análise e extração de informações baseadas em dados.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Em seguida realizei a extração dos arquivos enviados para o ambiente virtual “Jupyter”, facilitando a manipulação e transformação das informações.

```
#Loading DataFrames
contato=pd.read_csv(...\contacts.tsv", sep = "\t", usecols=[' contactsId', 'contactsName'])
compania=pd.read_csv(...\companies.tsv", sep = "\t", usecols=[0, 1, 6, 7, 9])
venda = pd.read_csv(...\deals.tsv", sep = "\t")
setor = pd.read_csv(...\sectors.tsv", sep = "\t")
```

Com os *DataFrames* já carregados, comecei as transformações para alcançar os outputs exigidos pela empresa. Modificando a nomenclatura de algumas colunas para facilitar a utilização das mesmas no futuro e juntando os dados em apenas uma tabela, unificada, onde as informações estavam de maneira mais simples de alcançar.

```
#Modifing Columns Names;
contato.columns = ['contactsId', 'contactsName']
compania.columns = ['companiesId', 'companiesName', 'contactsId', 'contactsName','sectorKey']
#Creating and Merging the Final DateFrame;
df_final = contato.copy()
df_final = df_final.merge(compania, on=['contactsId', 'contactsName'])
df_final = df_final.merge(venda, on=['contactsId', 'companiesId'])
print('Numero de linhas excluidas= ' + str(df_final.isnull().sum().sum()))
df_final.dropna(axis=0)

df_final.head(5)
```

Dando sequência ao desafio, após uma analise do conjunto gerado, foi necessário alterar os valores presentes em duas colunas para facilitar a utilização, no momento de traçar os gráficos. Para isso foi feito um loop, iterando sobre cada elemento da coluna, transforando-os, atribuindo o novo valor a uma lista criada e assim que finalizado substituindo todos os valores de uma única vez na coluna correspondente.

```
#Transforming Dates and Sectors
from datetime import datetime

setores = []
dates = []
for item in df_final['dealsDateCreated']:
    item  = str(datetime.strptime(item, "%m/%d/%Y"))
    item2 = item[:7]
```

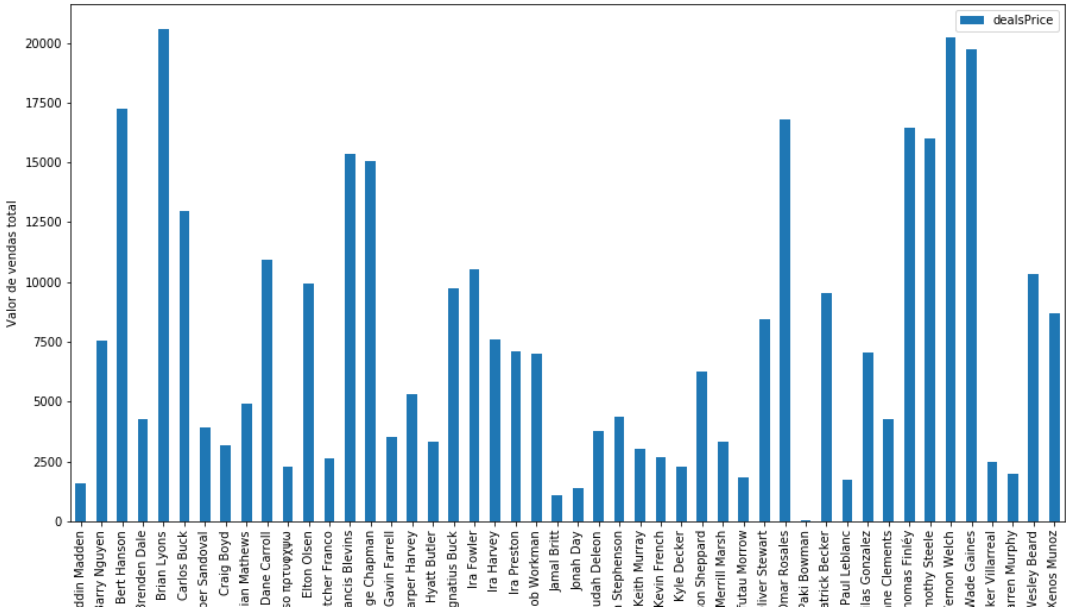
```
    dates.append(item2)

for i in df_final['sectorKey']:
    setores.append(setor['sector'][i-1])

df_final['dealsDateCreated'] = dates
df_final['sectorKey'] = setores

df_final.head(5)
```

	contactsId	contactsName	companiesId	companiesName	sectorKey	dealsId	dealsDateCreated	dealsPrice
0	1	Damian Mathews	1	Class LLP	Varejo	43	2019-02	660
1	1	Damian Mathews	1	Class LLP	Varejo	28	2019-05	60
2	1	Damian Mathews	1	Class LLP	Varejo	22	2017-11	4210
3	2	Paul Leblanc	2	Vulputate Corporation	Tecnologia	88	2018-04	1740
4	3	Thomas Finléy	3	Sed Dui Fusce Consulting	Atacado	83	2018-06	9590

Com todas as transformações finalizadas foi possível iniciar o processo de traçar os gráficos e salvar os *outputs* gerados. Novos *DataFrames* foram criados, com objetivo de agrupar apenas as colunas de interesse em função do resultado esperado.

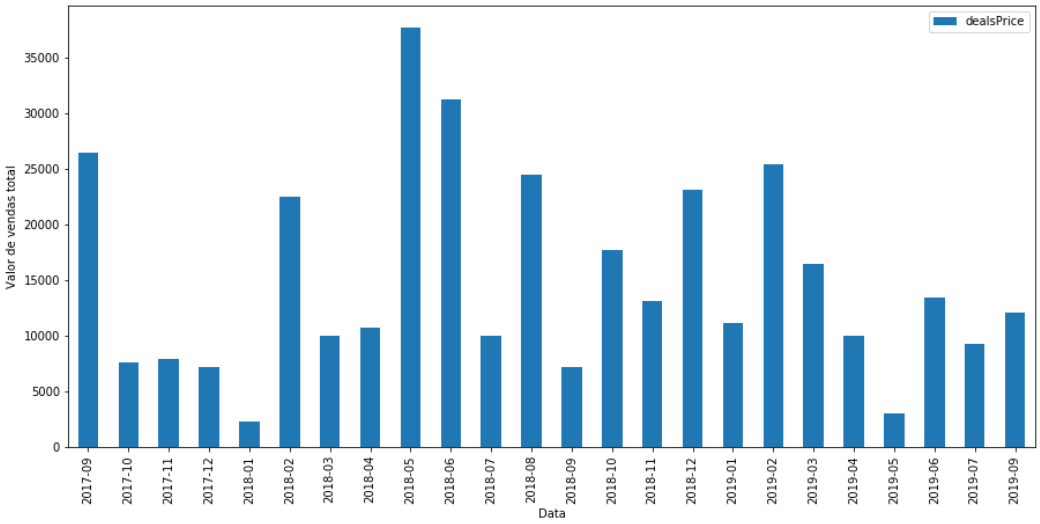
O primeiro gráfico gerado foi o de Valor Vendido por Contato (Vendedor), onde a variável agrupada foi o nome dos vendedores e a partir disso somou-se o valor total das vendas de cada indivíduo.

```
output1 = df_final[['contactsName', 'dealsPrice']]
output1 = output1.groupby(output1['contactsName']).sum()

#Plotting results 1;
fig, ax = plt.subplots(figsize=(15,8))
output1.plot(ax=ax, kind='bar')
plt.xlabel('Vendedor')
plt.ylabel('Valor vendido')
plt.savefig('graph_1.png')
plt.show()
```



O segundo gráfico gerado foi o de Valor Vendido por Mês, onde as datas foram classificadas de menos a mais recente, agrupadas e somadas o total de vendas de cada mês.

```
output2 = df_final[['dealsDateCreated', 'dealsPrice']]
output2['dealsDateCreated'] = pd.to_datetime(output2['dealsDateCreated'], format='%Y-%m').dt.strftime('%Y-%m')
output2.sort_values(['dealsDateCreated'], inplace=True)
output2 = output2.groupby(output2['dealsDateCreated']).sum()

#Plotting Result 2;
fig, ax = plt.subplots(figsize=(15,7))
output2.plot(ax=ax, kind='bar')
plt.xlabel('Data')
plt.ylabel('Valor de vendas total')
plt.savefig('graph_2.png')
plt.show()
```



Para o *output* final, a porcentagem de vendas de cada empresa por setor, foi necessario agrupar o conjunto de dados por duas variáveis, nome da companhia e setor em que esta atua, em seguida somar todos os valores relacionados a companhia e dividi-los pelo valor respectivo de cada setor.

```
output3 = df_final.groupby(['companiesName', 'sectorKey']).agg({'dealsPrice': 'sum'})
output3 = output3.groupby(level=0).apply(lambda x: 100 * x / float(x.sum()))

output3.to_csv('output_2.csv')
```