



Projeto Final da Disciplina Infraestrutura Google BigQuery – Pós Graduação MIT em Engenharia de dados: Big Data

Rafael Diniz Ramos

Google BigQuery

É um serviço de Data Warehouse para processamento de grandes volumes de dados, através de queries de consultas.

Conceitos Fundamentais

Armazenamento Colunar -> o armazenamento dos dados é colunar, deixando assim as consultas assertivas.

Cobrança por consulta -> a cobrança do serviço é feita através da quantidade de dados que será analisado/processado.

SQL -> o BigQuery usa linguagem SQL para fazer suas consultas.

Integração -> o BigQuery pode ser integrado com outras ferramentas do GCP.

Streaming de Dados -> é possível fazer análise de dados em tempos real.

Arquitetura do BigQuery

Como vimos acima em conceitos fundamentais, o BigQuery possui armazenamento colunar, ou seja, dados armazenados por colunas separadas e de forma compactada. Isso faz com que as consultas sejam rápidas possibilitando uma consulta com filtragem.

Por baixo do BigQuery, existe um sistema de arquivos distribuídos da Google, o Colossus. Ele armazena e replica os dados em vários servidores para que garanta assim a disponibilidade.

Além disso, os dados são separados em blocos e replicados em vários nós, para que aja paralelismo e tolerância a falhas.

Para fazer a consulta, o BigQuery utiliza o Dremel, que é o sistema de consulta do Google que utiliza SQL de forma distribuída.

Integração SQL e NoSQL

Apesar do BigQuery ser um serviço de Data Warehouse para análise e processamento de bases de dados em SQL, ele integra com alguns bancos NoSQL como o Firestore e o nativo Google Cloud Bigtable.

O Firestore é um banco NoSQL onde os arquivos tem o formato JSON, formato esse que é compatível com o BQ.

Outros que também são possíveis, é o Cassandra e o MongoDB, porém, não possui conexão direta com o BQ, os dados têm que ser exportados e salvos em CSV ou JSON para serem utilizados.

Em bases SQL, o BigQuery pode ser utilizado para fazer consultas de uma grande quantidade de dados, fazer análises e integração com ferramentas de visualização como o Looker Studio e Tableau.

Benefícios do BigQuery

Linguagem SQL -> utiliza SQL padrão p fazer consultas complexas eficientes.

Cobrança -> a cobrança é de acordo com a quantidade de processamento de dados da consulta.

Serverless -> não precisa de servidor, não sendo necessário infraestrutura e gerenciamento do hardware.

Desempenho -> as consultas são extremamente rápidas mesmo em grandes volumes de dados; pois sua estrutura de armazenamento colunar e o paralelismo nas consultas fazem com que a resposta seja rápida e eficiente.

Segurança -> oferece segurança avançada com criptografia e auditoria de acesso.

Pipeline de Dados

Pipeline de dados é uma infraestrutura, um tunelamento, com diversas tecnologias empregadas, para fazer a extração do dado, o tratamento e o seu carregamento (ETL).

Em meu dia a dia, o pipeline de dados começaria no nosso sistema ERP da empresa, onde são registradas as informações por exemplo de compra e venda. Eu extraio essas informações para um CSV com seus dados brutos, faço o tratamento, limpeza e enriquecimento com Python/Pandas e utilizo o Power BI para fazer análise


exploratória de dados e criação de dashboard, para assim gerar insights de compra e venda e apresentar para a Diretoria.

Prática

Vamos agora “subir” um dataframe para o BQ, mas antes vamos fazer alguns tratamentos em nossa base de dados.












Para isso iremos utilizar o Jupyter Notebook e Python.

Abaixo vamos visualizar nosso df.

 jupyter

ProjetoBigQuery (unsaved changes)

FileEditViewInsertCellKernelWidgetsHelp



Code

In [1]:

import pandas as pd

In [56]:

df = pd.read_csv('wc2018.csv', sep=",")
df.head(5)

Out[56]:

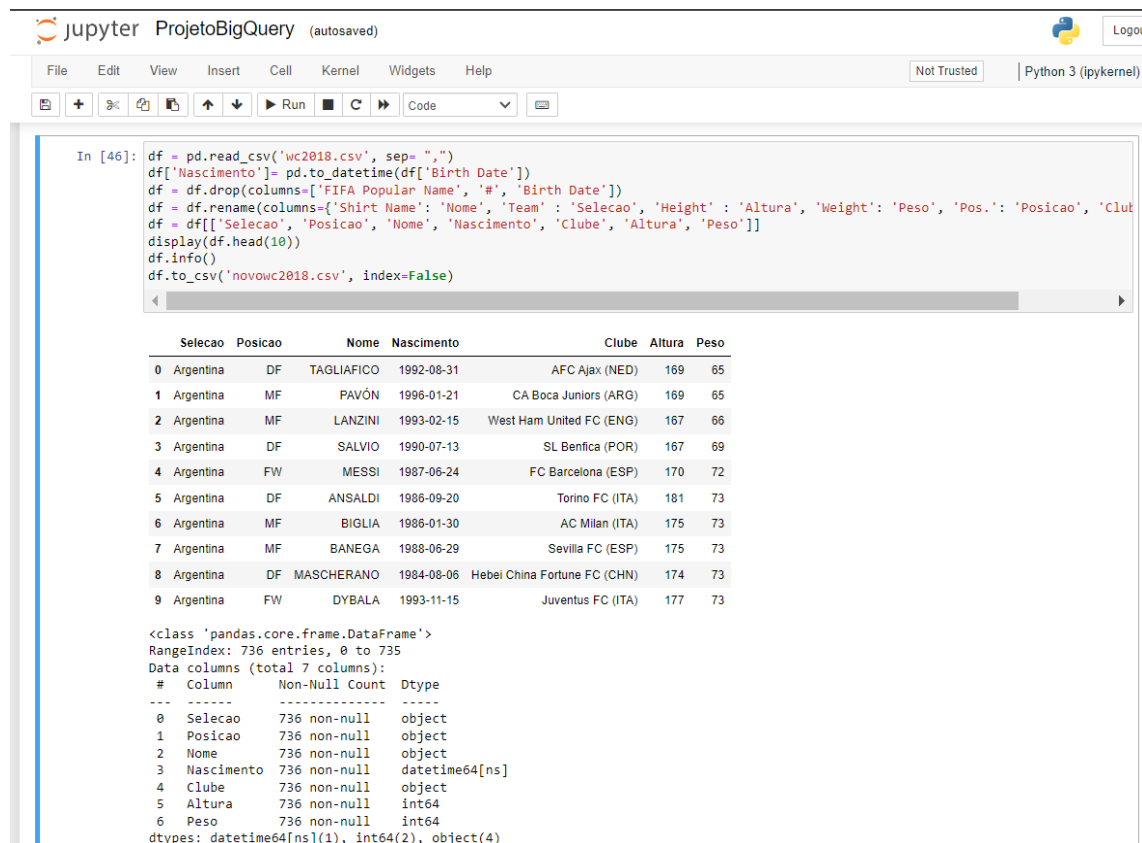
	Team	#	Pos.	FIFA Popular Name	Birth Date	Shirt Name	Club	Height	Weight
0	Argentina	3	DF	TAGLIAFICO Nicolas	31.08.1992	TAGLIAFICO	AFC Ajax (NED)	169	65
1	Argentina	22	MF	PAVON Cristian	21.01.1996	PAVÓN	CA Boca Juniors (ARG)	169	65
2	Argentina	15	MF	LANZINI Manuel	15.02.1993	LANZINI	West Ham United FC (ENG)	167	66
3	Argentina	18	DF	SALVIO Eduardo	13.07.1990	SALVIO	SL Benfica (POR)	167	69
4	Argentina	10	FW	MESSI Lionel	24.06.1987	MESSI	FC Barcelona (ESP)	170	72

In [55]:

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 736 entries, 0 to 735
Data columns (total 9 columns):
Column Non-Null Count Dtype
--- ---
0 Team 736 non-null object
1 # 736 non-null int64
2 Pos. 736 non-null object
3 FIFA Popular Name 736 non-null object
4 Birth Date 736 non-null object
5 Shirt Name 736 non-null object
6 Club 736 non-null object
7 Height 736 non-null int64
8 Weight 736 non-null int64
dtypes: int64(3), object(6)
memory usage: 51.9+ KB

Abaixo vamos excluir algumas colunas que não nos interessam para nossa análise, renomear as colunas, mudar o tipo de dado e trazer algumas informações do nosso dataframe.



```
In [46]: df = pd.read_csv('wc2018.csv', sep=",")
df['Nascimento'] = pd.to_datetime(df['Birth Date'])
df = df.drop(columns=['FIFA Popular Name', '#', 'Birth Date'])
df = df.rename(columns={'Shirt Name': 'Nome', 'Team': 'Selecao', 'Height': 'Altura', 'Weight': 'Peso', 'Pos.': 'Posicao', 'Club': 'Clube'})
df = df[['Selecao', 'Posicao', 'Nome', 'Nascimento', 'Clube', 'Altura', 'Peso']]
display(df.head(10))
df.info()
df.to_csv('novowc2018.csv', index=False)
```

	Selecao	Posicao	Nome	Nascimento	Clube	Altura	Peso
0	Argentina	DF	TAGLIAFICO	1992-08-31	AFC Ajax (NED)	169	65
1	Argentina	MF	PAVÓN	1996-01-21	CA Boca Juniors (ARG)	169	65
2	Argentina	MF	LANZINI	1993-02-15	West Ham United FC (ENG)	167	66
3	Argentina	DF	SALVIO	1990-07-13	SL Benfica (POR)	167	69
4	Argentina	FW	MESSI	1987-06-24	FC Barcelona (ESP)	170	72
5	Argentina	DF	ANSALDI	1986-09-20	Torino FC (ITA)	181	73
6	Argentina	MF	BIGLIA	1986-01-30	AC Milan (ITA)	175	73
7	Argentina	MF	BANEGA	1988-06-29	Sevilla FC (ESP)	175	73
8	Argentina	DF	MASCHERANO	1984-08-06	Hebei China Fortune FC (CHN)	174	73
9	Argentina	FW	DYBALA	1993-11-15	Juventus FC (ITA)	177	73

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 736 entries, 0 to 735
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Selecao     736 non-null   object
1   Posicao      736 non-null   object
2   Nome        736 non-null   object
3   Nascimento  736 non-null   datetime64[ns]
4   Clube       736 non-null   object
5   Altura      736 non-null   int64
6   Peso        736 non-null   int64
dtypes: datetime64[ns](1), int64(2), object(4)
```

Fazendo o upload de nossa base dados para o BigQuery.

Ao logar no Google Cloud, escolhemos a opção do menu BigQuery, BigQuery Studio.

Após, vamos em My First Project e criamos um novo projeto. O nome do meu projeto dado pelo Google foi mystic-rigging-417723. A hierarquia do BQ é Projeto, Conjunto de Dados e Tabela. É importante saber sobre essa hierarquia, pois ao dar o acesso de um projeto a um terceiro, você poderá restringir o nível de privilégio dado.

Com esta tela aberta, iremos clicar em adicionar, arquivo local.

Na origem vamos escolher “fazer upload” e selecionar o arquivo.

No destino, vamos criar primeiramente um conjunto de dados, onde nele iremos fazer a criação da tabela.

Em esquema, iremos marcar para identificar automaticamente, para que pegue o header do arquivo.

Visualização de nosso dataframe.

tabela_wcup18

CONSULTA

COMPARTILHAR

COPIAR

SNAPSHOT

EXCLUIR

EXPORTAR

ESQUEMA		DETALHES	VISUALIZAÇÃO	LINHAGEM	PERFIL DE DADOS		QUALIDADE DOS DADOS	
Linha	Selecao	Posicao	Nome	Nascimento	Clube	Altura	Peso	
1	Argentina	DF	TAGLIAFICO	1992-08-31	AFC Ajax (NED)	169	65	
2	Argentina	DF	SALVIO	1990-07-13	SL Benfica (POR)	167	69	
3	Argentina	DF	ANSALDI	1986-09-20	Torino FC (ITA)	181	73	
4	Argentina	DF	MASCHERANO	1984-08-06	Hebei China Fortune FC (CHN)	174	73	
5	Argentina	DF	ACUÑA	1991-10-28	Sporting CP (POR)	172	77	
6	Argentina	DF	MERCADO	1987-03-18	Sevilla FC (ESP)	181	81	
7	Argentina	DF	OTAMENDI	1988-12-02	Manchester City FC (ENG)	181	81	
8	Argentina	DF	ROJO	1990-03-20	Manchester United FC (ENG)	189	82	
9	Argentina	DF	FAZIO	1987-03-17	AS Roma (ITA)	199	85	
10	Australia	DF	BEHICH	1990-12-16	Bursaspor (TUR)	170	63	
11	Australia	DF	RISDON	1992-07-27	WS Wanderers FC (AUS)	169	70	
12	Australia	DF	MEREDITH	1988-05-04	Millwall FC (ENG)	179	71	
13	Australia	DF	SAINSBURY	1992-05-01	Grasshopper Club (SUI)	183	76	
14	Australia	DF	MILLIGAN	1985-04-08	Al Ahli SC (KSA)	178	78	
15	Australia	DF	JURMAN	1989-08-12	Suwon Samsung Bluewings FC...	190	83	
16	Australia	DF	DEGENEK	1994-04-28	Yokohama F-Marinos (JPN)	187	85	
17	Belgium	DF	DENDONCKER	1995-04-15	RSC Anderlecht (BEL)	188	76	

Resultados por página: 501 – 50 de 736

Estamos trabalhando um dataframe sobre a Copa do Mundo de 2018; vamos buscar algumas informações a respeito.

Quais jogadores do Brasil jogaram a Copa de 2018 e por quais clubes atuavam?

Sem título 5

EXECUTAR

SALVAR

FAZER O DOWNLOAD

COMPARTILHAR

```
1 SELECT Selecao, Nome, Clube
2 FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18`
3 WHERE
4 Selecao = 'Brazil'
```

Resultados da consulta

INFORMAÇÕES DO JOB	RESULTADOS	GRÁFICO	JSON	DETALHES DA EXECUÇÃO	GR
Linha	Selecao	Nome	Clube		
1	Brazil	FAGNER	SC Corinthians (BRA)		
2	Brazil	FILIFE LUIS	Atletico Madrid (ESP)		
3	Brazil	MARQUINHOS	Paris Saint-Germain FC (FRA)		
4	Brazil	MIRANDA	FC Internazionale (ITA)		
5	Brazil	DANILO	Manchester City FC (ENG)		
6	Brazil	T. SILVA	Paris Saint-Germain FC (FRA)		
7	Brazil	MARCELO	Real Madrid CF (ESP)		
8	Brazil	GEROMEL	Grêmio FBPA (BRA)		
9	Brazil	TAISON	FC Shakhtar Donetsk (UKR)		
10	Brazil	NEYMAR JR	Paris Saint-Germain FC (FRA)		
11	Brazil	D. COSTA	Juventus FC (ITA)		
12	Brazil	G. JESUS	Manchester City FC (ENG)		
13	Brazil	FIRMINO	Liverpool FC (ENG)		
14	Brazil	EDERSON	Manchester City FC (ENG)		
15	Brazil	A. BECKER	AS Roma (ITA)		
16	Brazil	CASSIO	SC Corinthians (BRA)		
17	Brazil	FRED	FC Shakhtar Donetsk (UKR)		
18	Brazil	FERNANDINHO	Manchester City FC (ENG)		
19	Brazil	P. COUTINHO	FC Barcelona (ESP)		
20	Brazil	WILLIAN	Chelsea FC (ENG)		
21	Brazil	PAULINHO	FC Barcelona (ESP)		
22	Brazil	CASEMIRO	Real Madrid CF (ESP)		
23	Brazil	R. AUGUSTO	Beijing Guoan (CHN)		

Quais jogadores das seleções da Copa jogavam no Barcelona?

Sem título 5

EXECUTAR

SALVAR

FAZER O DOWNLOAD

COMPARTILHAR

```
1 SELECT Seleccion, Nome, Clube
2 FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18`
3 WHERE
4 Clube = 'FC Barcelona (ESP)'
```

Resultados da consulta

INFORMAÇÕES DO JOB

RESULTADOS

GRÁFICO

JSON

DETALHES DA EXECUÇÃO

GRÁFICO

Linha	Selecção	Nome	Clube
1	Belgium	VERMAELEN	FC Barcelona (ESP)
2	Colombia	Y. MINA	FC Barcelona (ESP)
3	France	UMTITI	FC Barcelona (ESP)
4	Spain	JORDI ALBA	FC Barcelona (ESP)
5	Spain	PIQUÉ	FC Barcelona (ESP)
6	Argentina	MESSI	FC Barcelona (ESP)
7	France	DEMBELE	FC Barcelona (ESP)
8	Uruguay	SUAREZ	FC Barcelona (ESP)
9	Germany	TER STEGEN	FC Barcelona (ESP)
10	Brazil	P. COUTINHO	FC Barcelona (ESP)
11	Brazil	PAULINHO	FC Barcelona (ESP)
12	Croatia	RAKITIĆ	FC Barcelona (ESP)
13	Spain	A. INIESTA	FC Barcelona (ESP)
14	Spain	SERGIO	FC Barcelona (ESP)

Sem título 5 EXECUTAR SALVAR FAZER O DOWNLOAD COMPARTILHAR

```

1 SELECT t.Selecao, t.Nome, t.Altura AS altura_maxima
2 FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18` t
3 JOIN (
4     SELECT Selecao, MAX(Altura) AS altura_Maxima
5     FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18`
6     GROUP BY Selecao
7 ) max_altura
8 ON t.Selecao = max_altura.Selecao AND t.Altura = max_altura.altura_Maxima;
    
```

Resultados da consulta

INFORMAÇÕES DO JOB		RESULTADOS	GRÁFICO	JSON	DETALHES DA EXECUÇÃO
Linha	Selecao	Nome		altura_maxima	
1	Argentina	FAZIO		199	
2	Colombia	Y. MINA		194	
3	Costa Rica	K. WASTON		196	
4	Denmark	VESTERGAARD		200	
5	Egypt	A. HEGAZY		194	
6	Germany	SÜLE		195	
7	Japan	YOSHIDA		189	
8	Mexico	D. REYES		190	
9	Morocco	SAISS		190	

INFORMAÇÕES DO JOB

RESULTADOS

GRÁFICO

JSON

DETALHES

Linha	Selecão	Nome	altura_maxima
10	Portugal	FORTE	191
11	Serbia	MILENKOVIĆ	195
12	Spain	PIQUÉ	194
13	Switzerland	DJOUROU	192
14	Uruguay	COATES	196
15	Korea Republic	S W KIM	197
16	Morocco	BOUTAIB	190
17	Nigeria	NWANKWO	197
18	Russia	DZYUBA	196
19	Australia	JONES	193
20	Belgium	COURTOIS	199
21	Brazil	CASSIO	195
22	Croatia	L. KALINIĆ	201
23	England	BUTLAND	196
24	Iceland	SCHRAM	198
25	IR Iran	A. BEIRANVAND	194
26	Morocco	BOUNOU	190
27	Morocco	EL KAJOUI	190
28	Panama	RODRIGUEZ	197
29	Peru	GALLESE	189

30	Poland	SZCZESNY	195
31	Senegal	GOMIS	196
32	Serbia	STOJKOVIĆ	195
33	Sweden	OLSEN	198
34	Tunisia	BEN MUSTAPHA	192
35	France	NZONZI	197
36	Saudi Arabia	KANNO	192
37	Senegal	S. SANE	196

Quais os top 10 clubes que tiveram seus jogadores participando da Copa?

 Sem título 5

 EXECUTAR

 SALVAR ▾

 FAZER O DO

```
1 SELECT Clube, COUNT(*) AS total_jogadores
2 FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18`
3 GROUP BY Clube
4 ORDER BY total_jogadores DESC
5 LIMIT 10;
```

Resultados da consulta

INFORMAÇÕES DO JOB

RESULTADOS

GRÁFICO

JSON

Linha	Clube	total_jogadores
1	Manchester City FC (ENG)	16
2	Real Madrid CF (ESP)	15
3	FC Barcelona (ESP)	14
4	Chelsea FC (ENG)	12
5	Paris Saint-Germain FC (FRA)	12
6	Tottenham Hotspur FC (ENG)	12
7	FC Bayern München (GER)	11
8	Manchester United FC (ENG)	11
9	Juventus FC (ITA)	11
10	Atletico Madrid (ESP)	9

Mas antes, temos que salvar nossa consulta em forma de tabela, para podermos enviá-la para o Looker.

Adicionando a tabela com a consulta no Looker.

Adicionar dados ao relatório

Use o BigQuery BI Engine para que seus relatórios do BigQuery sejam carregados com mais rapidez.
[Ver mais](#)

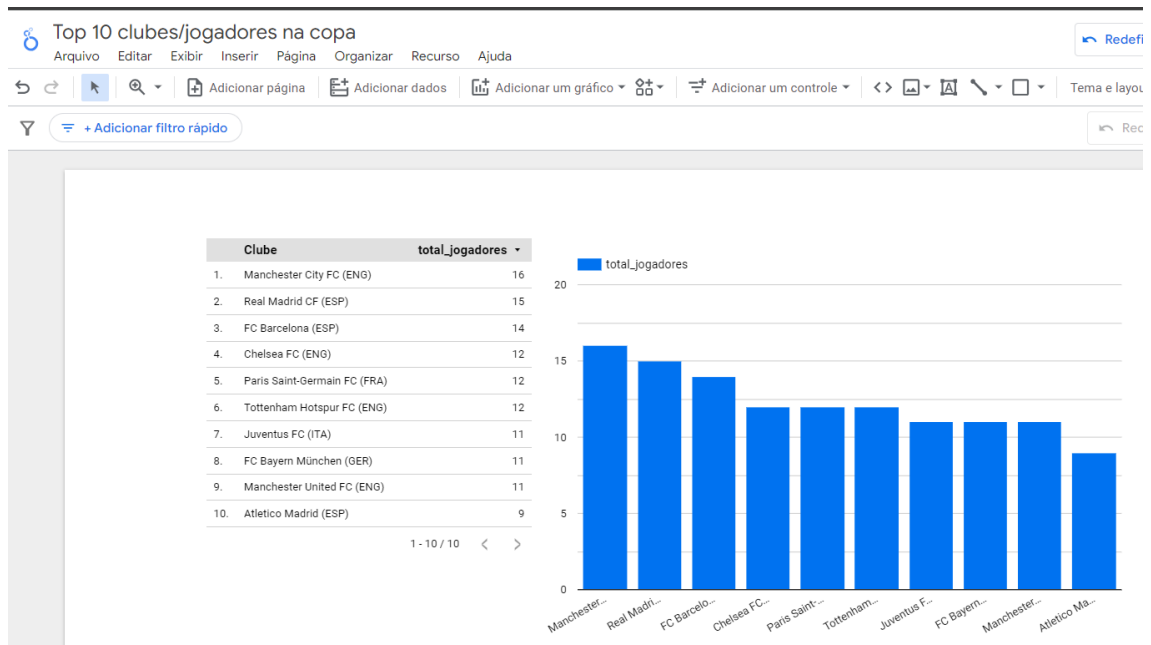
BigQuery

Por Google

O BigQuery é um serviço de armazenamento de dados de baixo custo, totalmente gerenciado e em escala de petabytes criado pelo Google para a análise de dados. A consulta e o processamento de dados realizados por esse serviço são pagos, e as cobranças são feitas no cartão de crédito registrado no projeto de faturamento.

[SAIBA MAIS](#)
[INFORMAR UM PROBLEMA](#)

PROJETOS RECENTES	Projeto	Conjunto de dados	Tabela
MEUS PROJETOS	Inserir o ID do projeto manualmente	bqml	tabela_wcup18
PROJETOS COMPARTILHADOS	My First Project	schema_csv	top10clubes
CONSULTA PERSONALIZADA	My First Project	schema_parquet	
CONJUNTOS DE DADOS PÚBLICOS	My First Project	schema_projeto	
	My First Project		



Vamos agora criar um gráfico de dispersão para compararmos duas variáveis, a de defensor (DF) com a altura referência de 1,88m.

O intuito é vermos se a maioria dos jogadores com 1,88m são defensores.

Para isso, iremos conectar o Colab ao BQ e migrarmos nosso dataframe inteiro pra lá.

Iremos usar novamente Python e as bibliotecas do Matplotlib e Seaborn.

Vamos instalar o Google Cloud, importar as bibliotecas supracitadas, importar o bigquery e a classe auth para nos autenticarmos no BQ.

Após, criamos um client onde por ele iremos conectar ao BQ para fazermos nossa consulta.

```
Projeto - BigQuery.ipynb ☆
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Código + Texto

[24] pip install google.cloud

Collecting google.cloud
  Downloading google_cloud-0.34.0-py2.py3-none-any.whl (1.8 kB)
Installing collected packages: google.cloud
Successfully installed google.cloud-0.34.0

[29] import matplotlib.pyplot as plt
import seaborn as sns

[2] from google.cloud import bigquery

[3] from google.colab import auth

[4] auth.authenticate_user()

[5] project_id = 'mystic-rigging-417723'

[6] client = bigquery.Client(project= project_id)
```

Aqui, criamos o objeto “dados” que recebe nossa tabela inteira através da instrução SQL e em seguida, transformamos em Dataframe. Após, vimos que 174 jogadores têm acima de 1,88m.

```
dados = client.query(''' SELECT * FROM `mystic-rigging-417723.schema_projeto.tabela_wcup18` ''')
df = dados.result().to_dataframe()

jogadores_acima_188 = df[df['Altura'] >= 188]
total_jogadores_acima_188 = len(jogadores_acima_188)
print(total_jogadores_acima_188)

174
```

Nesse passo, criamos a coluna Defensores através da coluna Posição e extraímos apenas os jogadores com a posição DF; em seguida, criamos a variável “df_filtrado” que recebeu da coluna Altura, jogadores com altura maior igual a 1,88m. Após, contamos quantas linhas atendem a condição e calculamos a porcentagem de cada.

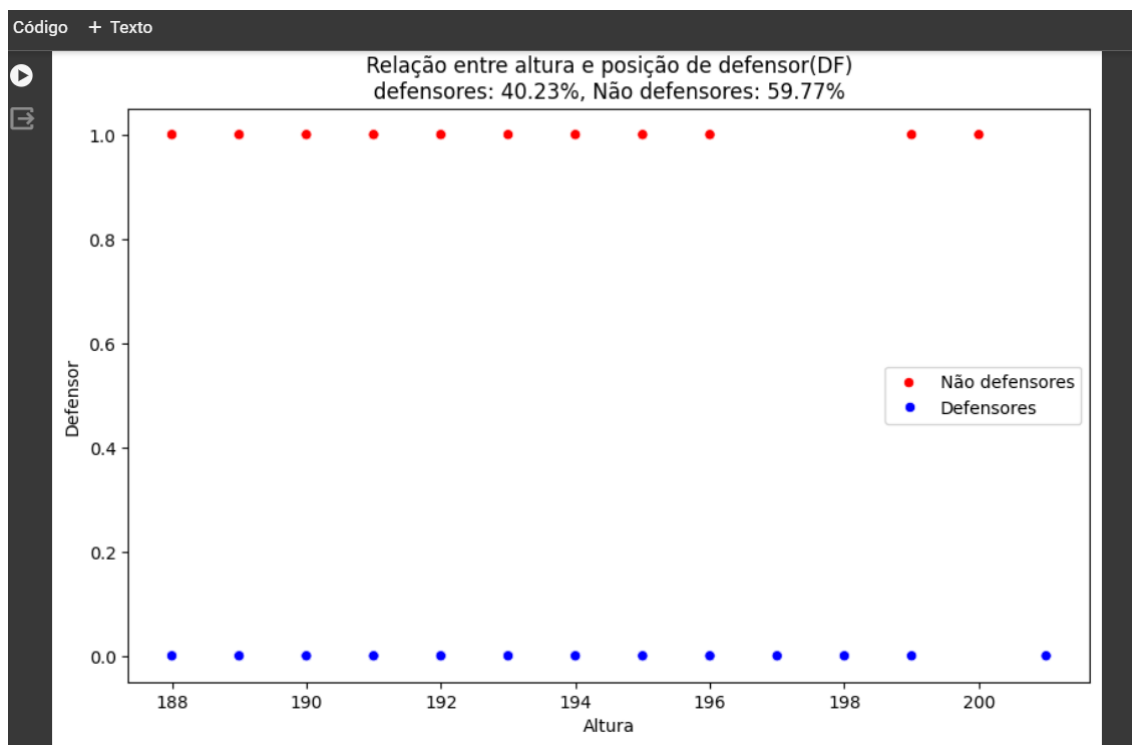
```
df['Defensores'] = df['Posicao'] == 'DF'

df_filtrado = df[df['Altura'] >= 188]

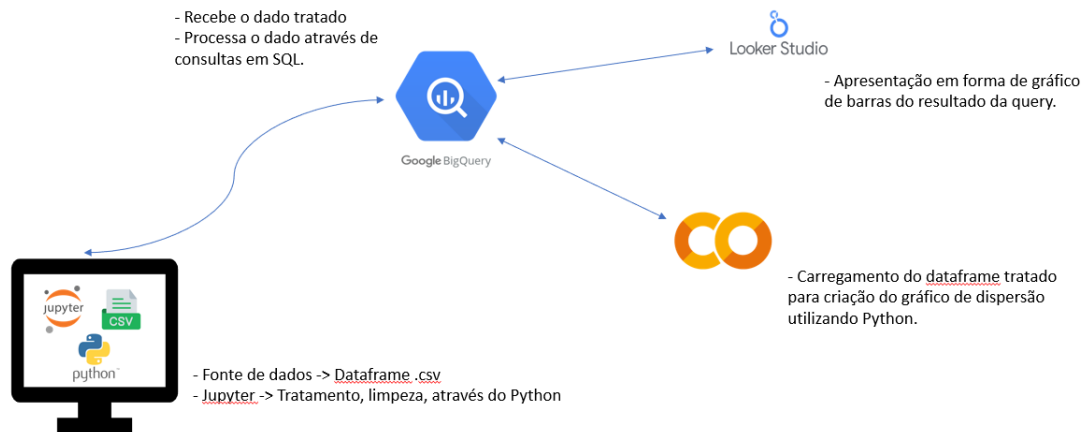
total = len(df_filtrado)
defensores = len(df_filtrado[df_filtrado['Defensores']])
nao_defensores = total - defensores
percent_defensores = (defensores / total) * 100
percent_nao_defensores = (nao_defensores / total) * 100

plt.figure(figsize=(10, 6))
sns.scatterplot(data=df_filtrado, x='Altura', y='Defensores', hue='Defensores', palette=[False: 'blue', True: 'red'])
plt.xlabel('Altura')
plt.ylabel('Defensor')
plt.title(f'Relação entre altura e posição de defensor(DF)\ndefensores: {percent_defensores:.2f}%, Não defensores: {percent_nao_defensores:.2f}%')
plt.legend(title='', loc='center right', labels=['Não defensores', 'Defensores'])
plt.show()
```

Agora podemos ver através do gráfico de dispersão que, surpreendentemente, a maioria dos jogadores com mais de 1,88m não são defensores.



Pipeline de dados do projeto



Considerações Finais

Primeiramente é bom dizer que é incrível conhecer essa ferramenta para consultas/análises/processamento de grandes volumes de dados!

Os resultados foram satisfatórios em minha opinião, e, poder agregar às outras ferramentas utilizadas nesse projeto, sem dúvida, enriquece e abre um leque para a criatividade e principalmente para o aprofundamento do conhecimento.

