ATP Players

Apresentação Semanal [3]

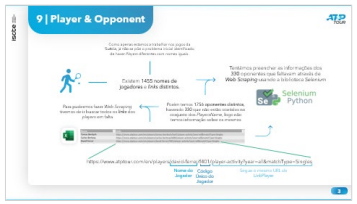
UC | Projeto Aplicado a Ciência de Dados I
Docentes | Diana Mendes **&** Sérgio Moro

Grupo 2

André Silvestre Nº104532
Diogo Catarino Nº104745
Francisco Gomes Nº104944
Rita Matos Nº104936

CDB1

Players & Opponents

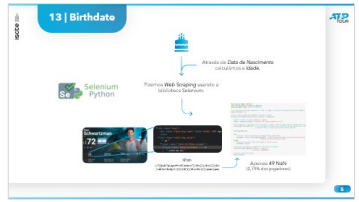


Hand

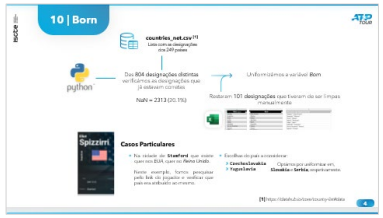
Já se encontra limpa.

Apresenta **1364 NaN** (11.85 %)

Birthdate



Born



Height

Apresenta **2292 NaN** (19.92 %).

Jogador com **15cm** de altura foi corrigido para **188cm** através de uma pesquisa no *Google*.

9 | Player & Opponent

Como apenas estamos a trabalhar nos jogos da **Suécia**, já **não** se põe o problema inicial identificado de haver *Players* diferentes com nomes iguais.

Existem **1455 nomes de jogadores** e ***links*** **distintos.**

Tentámos preencher as informações dos **330** oponentes que faltavam através de ***Web Scraping*** usando a biblioteca *Selenium*

Selenium Python

Porém temos **1756 oponentes distintos**, havendo **330** que não estão contidos no conjunto dos *PlayersName*, logo não temos informação sobre os mesmos

Para pudermos fazer *Web Scraping* tivemos de ir buscar todos os ***links*** dos *players* em falta

Nome	Links
Tomas Berdych	https://www.atptour.com/en/players/tomas-berdych/ba47/player-activity?year=all&matchType=Singles
Carlos Berlocq	https://www.atptour.com/en/players/carlos-berlocq/b884/player-activity?year=all&matchType=Singles
David Ferrer	https://www.atptour.com/en/players/david-ferrer/f401/player-activity?year=all&matchType=Singles

https://www.atptour.com/en/players/david-ferrer/f401/player-activity?year=all&matchType=Singles

Nome do Jogador

Código Único do Jogador

Segue o mesmo URL do LinkPlayer

10 | Born

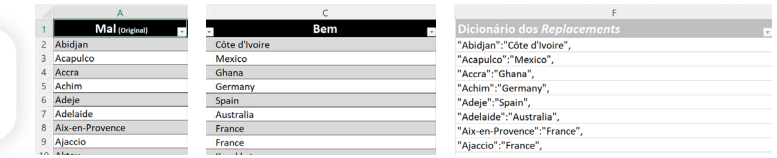
countries_net.csv [1]
Lista com as designações dos 249 países

Das **804 designações distintas** verificámos as designações que já estavam corretas

NaN = 2313 (20.1%)

Uniformizámos a variável *Born*

Restaram **101 designações** que tiveram de ser limpas manualmente



	A	C	F
1	Mal (Original)	Bem	Dicionário dos *Replacements*
2	Abidjan	Côte d'Ivoire	"Abidjan":"Côte d'Ivoire",
3	Acapulco	Mexico	"Acapulco":"Mexico",
4	Accra	Ghana	"Accra":"Ghana",
5	Achim	Germany	"Achim":"Germany",
6	Adeje	Spain	"Adeje":"Spain",
7	Adelaide	Australia	"Adelaide":"Australia",
8	Aix-en-Provence	France	"Aix-en-Provence":"France",
9	Ajaccio	France	"Ajaccio":"France",

Casos Particulares

- Na cidade de **`Stamford`** que existe quer nos *EUA*, quer no *Reino Unido*.

  Neste exemplo, fomos pesquisar pelo link do jogador e verificar que país era atribuído ao mesmo.

- Escolhas do país a considerar:
  - **`Czechoslovakia`**
  - **`Yuguslavia`**

  Optámos por uniformizar em, **`Slovakia`** e **`Serbia`**, respetivamente.

[1] *https://datahub.io/core/country-list#data*

13 | Birthdate

Através da *Data de Nascimento* calculámos a *Idade.*

Selenium Python

Fizemos *Web Scraping* usando a biblioteca *Selenium*.


```
<div class="wrap">
  <div class="table-big-label" style="width: 50%">Age
  </div>
  <div class="table-big-value">
    " 30 "
    <span class="table-birthday-wrapper">
      <span class="table-birthday"> == $0
        " (1992.08.16) "
```

XPath

```
//*[@id="playerProfileHero"]/div[2]/div[2]/div
/table/tbody/tr[1]/td[1]/div/div[2]/span/span
```

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from tqdm import tqdm # Barra de Progresso

# Cria uma lista vazia para armazenar os links, as idades e as datas de nascimento dos vários jogadores
player_birthdays = []

# Iterar sobre os nomes dos jogadores no dataset original
for atp_url in tqdm(list(atp_suecia['LinkPlayer'].unique())):

    # Abrir o Chrome e acessar a página do jogador
    driver = webdriver.Chrome("C:\Program Files (x86)\chromedriver.exe")

    # Para aumentar o tempo limite seria adicionar a seguinte linha de código antes da tentativa de enco
    driver.implicitly_wait(100)   # Isso irá definir um tempo limite implícito de 100 segundos para o We
                                  # aguardar antes de lançar uma exceção TimeoutException.

    driver.get(atp_url)

    try:
        # Extrair a data de nascimento do elemento HTML usando XPath
        birthdate = driver.find_element(By.XPATH, '//*[@id="playerProfileHero"]/div[2]/div[2]/div/table/
                                        'tbody/tr[1]/td[1]/div/div[2]').text
    except:
        # Caso não encontre a data de nascimento, atribui np.nan
        birthdate = np.nan

    # Adicionar informações do jogador na lista como um dicionário
    player_birthdays.append({'LinkPlayer': atp_url,
                             'Birthdate': birthdate})

    # Fechar a janela do Chrome
    driver.close()
```

Apenas **49 NaN**
(2,74% dos jogadores)

Fases da Limpeza

Duplicados

Na base de dados começámos por eliminar os jogos duplicados.

Assim, retirámos desde logo todas as linhas duplicadas, sendo eliminadas **2830 obs.** (0,22%)

01

Jogadores e Jogos

Limpámos os jogos e jogadores, restringindo apenas à Suécia, restando assim **11 508 obs.**

Adicionalmente, retirámos ainda as observações com jogos em torneios de duplas.

02

Jogos Únicos

Nesta fase retirámos os jogos espelhados, obtendo-se:

5994 observações (52.1%)

03