

INFNET

Desenvolvimento Front-End com Python (com Streamlit) [24E3_1] - TP3

Análise exploratória de Dados de Turismo no Rio de Janeiro

Tabela 2675 - [Chegada mensal de turistas pelo Rio de Janeiro, por via Aérea, segundo continentes e países de residência permanente, entre 2006-2019](#)

Link GitHub: <https://github.com/r-moreira/infnet-streamlit-tp3>

1. **Objetivo:** Realizar uma análise exploratória de dados sobre a chegada mensal de turistas no Rio de Janeiro, por via aérea, segundo continentes e países de residência permanente, entre 2006-2019.
2. **Motivação:** O turismo é uma das principais atividades econômicas do Rio de Janeiro, sendo responsável por grande parte da movimentação financeira da cidade. A análise dos dados de turismo é fundamental para entender o comportamento dos turistas e auxiliar na tomada de decisões estratégicas para o setor.

```
In [1]: import pandas as pd

class DataRioTableParser:
    def __init__(self):
        self.table_2675_columns = [
            'Continentes e países de residência permanente',
            'Total',
            'Janeiro',
            'Fevereiro',
            'Março',
            'Abril',
            'Maio',
            'Junho',
            'Julho',
            'Agosto',
            'Setembro',
            'Outubro',
            'Novembro',
            'Dezembro',
            'year'
        ]
        self.table_2675_continents = [
            'África',
            'América Central',
            'América Central e Caribe',
```

```

        'América do Norte',
        'América do Sul',
        'Ásia',
        'Europa',
        'Oceania'
    ]

def _convert_xls_to_dataframe(self, xls_file_path: str) -> pd.DataFrame:
    xls = pd.ExcelFile(xls_file_path, engine='xlrd')
    all_sheets_df = pd.DataFrame()
    for sheet_name in xls.sheet_names:
        df = pd.read_excel(xls, sheet_name=sheet_name)
        df['year'] = sheet_name
        all_sheets_df = pd.concat([all_sheets_df, df], ignore_index=True)
    return all_sheets_df

def _clean_and_separate_data(self, df: pd.DataFrame) -> tuple[pd.DataFrame, pd.DataFrame]:
    df.columns = self.table_2675_columns

    df.dropna(how='all', inplace=True)
    df = df[df.apply(lambda x: len(x.dropna()) == len(df.columns), axis=1)]
    df = df[df['Continentes e países de residência permanente'].str.strip() != '']
    df.reset_index(drop=True, inplace=True)

    continents = []
    countries = []
    current_continent = None

    for _, row in df.iterrows():
        if pd.isna(row['Continentes e países de residência permanente']):
            continue
        if any(row['Continentes e países de residência permanente'].str.strip() != ''):
            current_continent = row['Continentes e países de residência permanente'].str.strip()
            continents.append({'Continente': current_continent, **row.to_dict()})
        else:
            countries.append({'Continente': current_continent, **row.to_dict()})

    return pd.DataFrame(continents), pd.DataFrame(countries)

def _clean_final_dataframes(self, continents_df: pd.DataFrame, countries_df: pd.DataFrame):
    continents_df.drop(columns=['Continentes e países de residência permanente'], inplace=True)
    continents_df.loc[continents_df['Continente'] == 'América Central', 'Continente'] = 'América Central'

    countries_df.rename(columns={'Continentes e países de residência permanente': 'País'}, inplace=True)
    countries_df['País'] = countries_df['País'].str.strip()
    countries_df.loc[countries_df['Continente'] == 'América Central', 'Continente'] = 'América Central'
    countries_df.replace({'-': 0, ' - ': 0}, inplace=True)

    return continents_df, countries_df

def get_from_table_2675(self, xls_file_path: str) -> tuple[pd.DataFrame, pd.DataFrame]:
    all_sheets_df = self._convert_xls_to_dataframe(xls_file_path)
    continents_df, countries_df = self._clean_and_separate_data(all_sheets_df)
    return self._clean_final_dataframes(continents_df, countries_df)

```

```
In [2]: import os

continents_df, countries_df = DataRiotTableParser().get_from_table_2675(xls_f
continents_df.to_csv('../data/02_processed/table_2675_continents.csv', index
countries_df.to_csv('../data/02_processed/table_2675_countries.csv', index=F
```

```
In [3]: df_continents = pd.read_csv('../data/02_processed/table_2675_continents.csv')
df_countries = pd.read_csv('../data/02_processed/table_2675_countries.csv')
```

```
In [4]: df_continents.head()
```

```
Out[4]:
```

	Continente	Total	Janeiro	Fevereiro	Março	Abril	Mai	Junho	Julho	Agos
0	África	29041	2436	1848	1927	1802	1529	1825	1882	456
1	América Central e Caribe	3546	267	487	292	327	131	341	365	19
2	América do Norte	185680	22308	21452	19147	16439	16175	17453	18185	763
3	América do Sul	122387	20200	17108	16375	9654	5217	7126	7645	608
4	Ásia	18125	2342	1475	1727	1094	1522	1420	1409	137

```
In [5]: df_continents.tail()
```

```
Out[5]:
```

	Continente	Total	Janeiro	Fevereiro	Março	Abril	Mai	Junho	Julho	Ag
92	América do Norte	138629	10570	12411	16720	8907	9198	10428	10719	1
93	América do Sul	627135	80206	75954	69888	46010	36775	43209	43584	4
94	Ásia	39320	2956	4300	5709	2692	3463	2583	2765	
95	Europa	357829	28503	39488	38102	31461	20849	18919	27624	3
96	Oceania	14694	1788	1697	2104	840	719	879	921	

```
In [6]: df_countries.head()
```

Out [6]:	Continente	País	Total	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho
0	África	África do Sul	3012	363	203	103	85	38	442	474
1	África	Angola	21606	1639	1113	1208	1501	940	966	1037
2	África	Cabo Verde	2407	221	286	212	103	301	326	274
3	África	Nigéria	233	22	7	35	0	18	8	8
4	África	Outros	1783	191	239	369	113	232	83	89

In [7]: `df_countries.tail()`

Out [7]:

	Continente	País	Total	Janeiro	Fevereiro	Março	Abril	Maio	Junho
924	Europa	Outros países da Europa	1649	158	139	281	90	96	66
925	Oceania	Austrália	12929	1706	1598	1958	650	573	740
926	Oceania	Nova Zelândia	1734	81	98	140	190	145	136
927	Oceania	Outros países da Oceania	31	1	1	6	0	1	3
928	Oceania	Países não especificados	7	2	0	1	0	0	1

In [8]: `df_countries.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 929 entries, 0 to 928
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Continente  929 non-null    object
1   País        929 non-null    object
2   Total       929 non-null    int64
3   Janeiro     929 non-null    int64
4   Fevereiro   929 non-null    int64
5   Março       929 non-null    int64
6   Abril       929 non-null    int64
7   Maio        929 non-null    int64
8   Junho       929 non-null    int64
9   Julho       929 non-null    int64
10  Agosto      929 non-null    int64
11  Setembro    929 non-null    int64
12  Outubro     929 non-null    int64
13  Novembro    929 non-null    int64
14  Dezembro    929 non-null    int64
15  year        929 non-null    int64
dtypes: int64(14), object(2)
memory usage: 116.3+ KB
```

```
In [9]: df_continents.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97 entries, 0 to 96
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Continente  97 non-null    object
1   Total       97 non-null    int64
2   Janeiro     97 non-null    int64
3   Fevereiro  97 non-null    int64
4   Março       97 non-null    int64
5   Abril       97 non-null    int64
6   Maio        97 non-null    int64
7   Junho       97 non-null    int64
8   Julho       97 non-null    int64
9   Agosto      97 non-null    int64
10  Setembro    97 non-null    int64
11  Outubro     97 non-null    int64
12  Novembro    97 non-null    int64
13  Dezembro    97 non-null    int64
14  year        97 non-null    int64
dtypes: int64(14), object(1)
memory usage: 11.5+ KB
```

```
In [10]: df_countries['País'].unique()
```

```
Out[10]: array(['África do Sul', 'Angola', 'Cabo Verde', 'Nigéria', 'Outros',
                'Costa Rica', 'Panamá', 'Porto Rico', 'Canadá', 'Estados Unidos',
                'México', 'Argentina', 'Bolívia', 'Chile', 'Colômbia', 'Equador',
                'Guiana Francesa', 'Paraguai', 'Peru', 'República da Guiana',
                'Suriname', 'Uruguai', 'Venezuela', 'China', 'Japão',
                'República da Coreia', 'Alemanha', 'Áustria', 'Bélgica',
                'Dinamarca', 'Espanha', 'Finlândia', 'França', 'Grécia', 'Holanda',
                'Hungria', 'Inglaterra', 'Irlanda', 'Itália', 'Noruega', 'Polônia',
                'Portugal', 'Suécia', 'Suíça', 'Austrália', 'Nova Zelândia',
                'Oriente Médio', 'Arábia Saudita', 'Irake', 'Israel',
                'Países não especificados', 'Cuba', 'Índia', 'República Tcheca',
                'Rússia', 'Egito', 'Gana', 'Marrocos', 'Moçambique', 'Quênia',
                'Tunísia', 'Outros países da África', 'El Salvador', 'Guatemala',
                'Haiti', 'Honduras', 'Nicarágua', 'República Dominicana',
                'Trinidad e Tobago', 'Outros países da América Central e Caribe',
                'Guiana', 'Bangladesh', 'China, Hong Kong', 'Cingapura',
                'Filipinas', 'Indonésia', 'Irã', 'Líbano', 'Malásia', 'Paquistão',
                'República da Coreia', 'Síria', 'Tailândia', 'Taiwan',
                'Outros países da Ásia', 'Bulgária', 'Croácia', 'Eslováquia',
                'Eslovênia', 'Estônia', 'Letônia', 'Lituânia', 'Luxemburgo',
                'Reino Unido', 'Romênia', 'Sérvia', 'Turquia', 'Ucrânia',
                'Outros países da Europa', 'Outros países da Oceania'],
              dtype=object)
```

```
In [11]: df_continents['Continente'].unique()
```

```
Out[11]: array(['África', 'América Central e Caribe', 'América do Norte',
                'América do Sul', 'Ásia', 'Europa', 'Oceania'], dtype=object)
```

```
In [13]: df_continents.columns
```

```
Out[13]: Index(['Continente', 'Total', 'Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio',  
              'Junho', 'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro',  
              'Dezembro', 'year'],  
             dtype='object')
```

```
In [19]: # Export para pdf  
!jupyter nbconvert --to webpdf rodrigo_avila_DR1_TP3.ipynb --allow-chromium-
```

```
[NbConvertApp] Converting notebook rodrigo_avila_DR1_TP3.ipynb to webpdf
```

```
[NbConvertApp] Building PDF
```

```
[NbConvertApp] PDF successfully created
```

```
[NbConvertApp] Writing 253033 bytes to ../rodrigo_avila_DR1_TP3.pdf.pdf
```