In [1]:

```
# In[0]:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import postgres_conn as pgc
import category_encoders as ce
```

In [2]:

```
# In[1]: Funcões
sns.set_style('darkgrid')

SQL = '''

select * from queimadas_brasil_reservas

'''
```

In [3]:

```
SQL2 = '''

WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    CASE
        WHEN a.bioma = 'Cerrado' THEN 0
        WHEN a.bioma = 'Mata Atlantica' THEN 1
        WHEN a.bioma = 'Pantanal' THEN 2
        WHEN a.bioma = 'Amazonia' THEN 3
        WHEN a.bioma = 'Pampa' THEN 4
        WHEN a.bioma = 'Caatinga' THEN 5
        ELSE 6
    END AS flag_bioma,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.estado)

'''
```

In [4]:

```
amazonia_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Amazonia'
'''
```

In [5]:

```
caatinga_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Caatinga'
'''
```

In [6]:

```
cerrado_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Cerrado'
'''
```

In [7]:

```
m_atlantica_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Mata Atlantica'
'''
```

In [8]:

```
pampa_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Pampa'
'''
```

In [9]:

```
pantanal_sql = '''
WITH meteor AS(
    SELECT
        CAST("Data" AS date) AS m_data,
        "uf" AS m_estado,
        AVG("PRECIPITAÇÃO_TOTAL,_HORÁRIO_(mm)") AS avg_prep_total,
        AVG("PRESSAO_ATMOSFERICA_AO_NIVEL_DA_ESTACAO,_HORARIA_(mB)") AS avg_pressao_atm,
        AVG("PRESSÃO_ATMOSFERICA_MAX.NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_max,
        AVG("PRESSÃO_ATMOSFERICA_MIN._NA_HORA_ANT._(AUT)_(mB)") AS avg_press_atm_min,
        AVG("RADIACAO_GLOBAL_(Kj/m²)") AS avg_rad_global,
        AVG("TEMPERATURA_DO_AR___BULBO_SECO,_HORARIA_(°C)") AS avg_temp_ar,
        AVG("UMIDADE_RELATIVA_DO_AR,_HORARIA_(%)") AS avg_umd_ar,
        AVG("VENTO,_VELOCIDADE_HORARIA_(m/s)") AS avg_vento_velo
    FROM metereologia
    GROUP BY "Data", "uf"
    ORDER BY "Data")

SELECT
    a.*,
    b.*
FROM queimadas_brasil_reservas AS a,
    meteor AS b
WHERE
    CAST(a.datahora AS date) = b.m_data
    AND trim(a.estado) = trim(b.m_estado)
    AND a.bioma = 'Pantanal'
'''
```

In [10]:

```
# In[1]: Funcões
def get_encod(df):

    target_encoder = ce.TargetEncoder(cols=['estado',
                                            'municipio',
                                            'bioma'], smoothing=0, return_df=True)

    df  = target_encoder.fit_transform(df, df['frp'])

    return df


def interpolate(df):

    df['riscofogo'] = df['riscofogo'].interpolate(method='linear')
    df['diasemchuv'] = df['diasemchuv'].interpolate(method='linear')
    df['avg_prep_total'] = df['avg_prep_total'].interpolate(method='linear')
    df['avg_pressao_atm'] = df['avg_pressao_atm'].interpolate(method='linear')
    df['avg_press_atm_max'] = df['avg_press_atm_max'].interpolate(method='linear')
    df['avg_press_atm_min'] = df['avg_press_atm_min'].interpolate(method='linear')
    df['avg_rad_global'] = df['avg_rad_global'].interpolate(method='linear')
    df['avg_temp_ar'] = df['avg_temp_ar'].interpolate(method='linear')
    df['avg_umd_ar'] = df['avg_umd_ar'].interpolate(method='linear')
    df['avg_vento_velo'] = df['avg_vento_velo'].interpolate(method='linear')

    return df

def create_var(df):

    df['datahora'] = pd.to_datetime(df['datahora'])
    df['data'] = pd.to_datetime(df['datahora']).dt.date
    df['hora'] = pd.to_datetime(df['datahora']).dt.hour
    df['minuto'] = pd.to_datetime(df['datahora']).dt.minute
    df['mes'] = pd.to_datetime(df['data']).dt.month
    df['quadrimestre'] = pd.to_datetime(df['data']).dt.quarter
    df['dia_da_semana'] = pd.to_datetime(df['data']).dt.dayofweek

    df['diasemchuv'] = df['diasemchuv'].apply(lambda x: None if x <= 0 else x)
    df['riscofogo'] = df['riscofogo'].apply(lambda x: None if x <= 0 else x)

    #df = interpolate(df)

    df = df.dropna(subset=['frp'])


    return df

def select_vars(df):
    col_names = ['frp','estado','m_estado','municipio',
                 'bioma','riscofogo','diasemchuv','superficie',
                 'avg_prep_total','avg_pressao_atm','avg_rad_global','avg_temp_ar',
                 'avg_umd_ar','avg_vento_velo','hora','minuto',
                 'mes','dia_da_semana', 'quadrimestre','flg_q_r']

    return df[col_names]

def barplot_queimadas(df):

    f, ax = plt.subplots(figsize=(15, 10))
    sns.barplot(x="bioma", y="frp", data=df)
    ax.set(title="Histograma Bioma")
    plt.show()

    f, ax = plt.subplots(figsize=(15, 10))
    sns.barplot(y="estado", x="frp", data=df)
    ax.set(title="Histograma Estado")
    plt.show()


    f, ax = plt.subplots(figsize=(15,10))
    sns.histplot(data=df, x=df['diasemchuv'],kde=True)
    ax.set(title="Dias sem Chuva")
    plt.show()

    return

def lineplot_queimadas(df):

    f, ax = plt.subplots(figsize=(15, 10))
    sns.lineplot(x=df.index, y="diasemchuv",label ='diasemchuv',data=df)
    plt.suptitle("Queimadas diasemchuv")
    plt.show()

    f, ax = plt.subplots(figsize=(15, 10))
    sns.lineplot(x=df.index, y="precipitac",label ='precipitac',data=df)
    plt.suptitle("Queimadas precipitac")
    plt.show()

    f, ax = plt.subplots(figsize=(15, 10))
    sns.lineplot(x=df.index, y="riscofogo",label ='riscofogo',data=df)
    plt.suptitle("Queimadas riscofogo")
    plt.show()

    f, ax = plt.subplots(figsize=(15, 10))
    sns.lineplot(x=df.index, y="frp",label ='frp',data=df)
    plt.suptitle("Queimadas frp")
    plt.show()

    return

def export_data(df):
    #df = select_vars(df_raw)
    df.to_csv('queimadas.csv',index=False)
    return
```

In [11]:

```
# In[2]: Conectar com Postgres SQL

pwd = input("Informe a senha do database: ")

conn = pgc.db_conn('localhost','spatial','postgres',pwd)
conn.postgres()

amazonia_raw = conn.get_dataframe(amazonia_sql)
caatinga__raw = conn.get_dataframe(caatinga_sql)
cerrado_raw = conn.get_dataframe(cerrado_sql)
m_atlantica_raw = conn.get_dataframe(m_atlantica_sql)
pampa_raw = conn.get_dataframe(pampa_sql)
pantanal_raw = conn.get_dataframe(pantanal_sql)
```

```
Informe a senha do database: 1234
```

In [12]:

```
# In[3]: Analise de dados
#queimadas = get_encod(queimadas_raw)
amazonia = select_vars(create_var(amazonia_raw))
amazonia_corr = amazonia.corr()

caatinga = select_vars(create_var(caatinga__raw))
caatinga_corr = caatinga.corr()

cerrado = select_vars(create_var(cerrado_raw))
cerrado_corr = cerrado.corr()

m_atlantica = select_vars(create_var(m_atlantica_raw))
m_atlantica_corr = m_atlantica.corr()

pampa = select_vars(create_var(pampa_raw))
pampa_corr = pampa.corr()


pantanal = select_vars(create_var(pantanal_raw))
pantanal_corr = pantanal.corr()
```

In [13]:

```
#na_df = queimadas.isnull().sum(axis = 0)

# In[3]: Analise de dados

#queimadas_corr = queimadas.corr()
#queimadas_describe = queimadas.describe()



#na_df = queimadas.isnull().sum(axis = 0)

# In[4]: Analise de dados
#lineplot_queimadas(queimadas)
#barplot_queimadas(queimadas)

# In[5]: Analise de dados
#export_data(queimadas)


#pantanal.to_csv('pantanal.csv',index=False)

#amazonia.to_csv('amazonia.csv',index=False)
```

In [ ]:

```

```