In [67]:

```
import pandas as pd
from utils.gerar_score import Score
```

In [68]:

```
df_features = pd.read_excel("C:\\Users\\DELL\\Google Drive\\2021-2\\TCC\\Dataset\\DADOS_SCORE_21_09_refatoracao.xlsx",sheet_name="Planilha2")
```

In [69]:

```
df_features_dh = pd.read_excel("C:\\Users\\DELL\\Google Drive\\2021-2\\TCC\\Dataset\\DADOS_SCORE_21_09_Pablo.xlsx",sheet_name="Planilha2")
```

In [70]:

```
df = pd.read_excel("C:\\Users\\DELL\\Google Drive\\2021-2\\TCC\\Dataset\\dados tratados missing 5.xlsx")
```

In [71]:

```
df.head()
```

Out[71]:

	IN002	IN031	IN101	IN049	IN019	IN023	IN024	IN0
0	573.250000	29.490000	129.760000	27.520000	505.210000	91.530000	14.560000	57.1900
1	396.040000	42.860000	92.390000	28.120000	268.670000	96.830000	3.470000	84.0900
2	248.830000	49.620000	102.380000	30.760000	162.330000	98.980000	98.980000	69.5400
3	532.220000	170.890000	26.440000	89.860000	453.060000	99.880000	99.420000	99.7000
4	365.250000	34.900000	102.620000	22.280000	319.540000	73.690000	15.010000	34.4600

5 rows × 41 columns

In [72]:

```
df_features.head()
```

Out[72]:

	Variavel	Grupo	Sentido
0	IN002	Eficiencia	0
1	Tarifa	Eficiencia	0
2	IN031	Eficiencia	1
3	IN101	Eficiencia	0
4	IN049	Eficiencia	1

In [73]:

```
df.Tarifa
```

Out[73]:

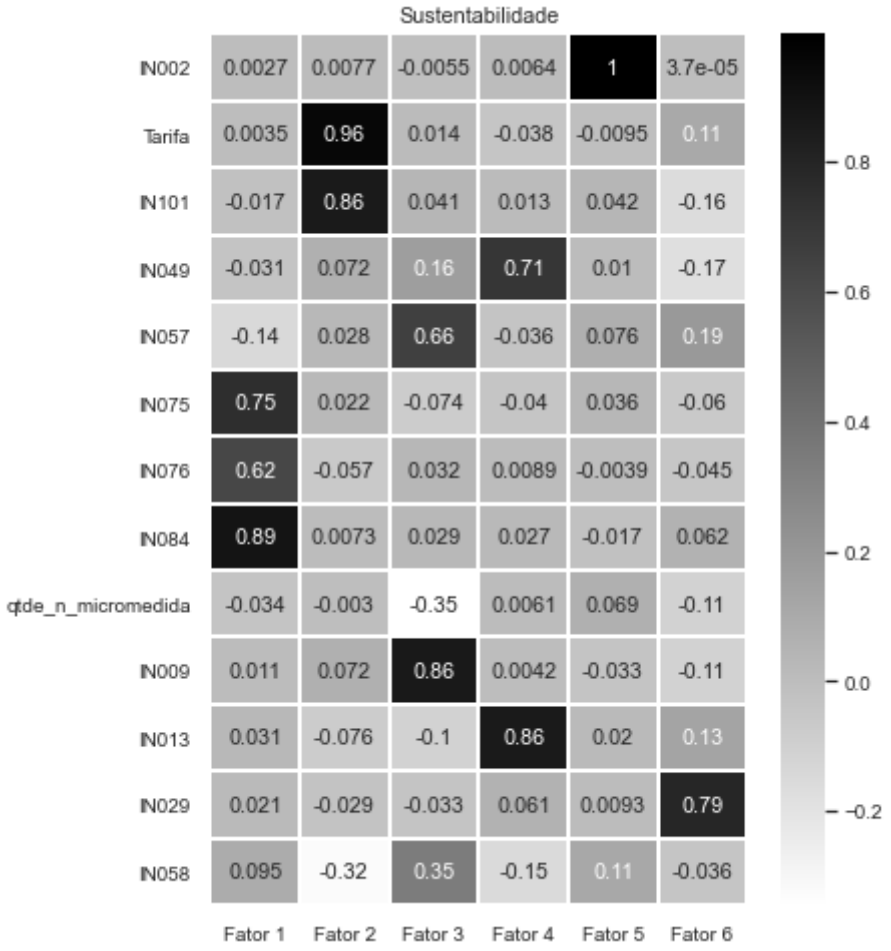
```
0      1.208817
1      0.890815
2      0.967672
3      0.252577
4      0.785196
         ...
807    0.859589
808    0.740630
809    0.996117
810    0.709877
811    0.892206
Name: Tarifa, Length: 812, dtype: float64
```

In [74]:

```
import os
if os.path.exists("Scores.xlsx") == True:
    os.remove("Scores.xlsx")
```

In [75]:

```
score=Score(df,df_features_dh,delta=0.05,normalizar_score=True,save_excel=True,rotacao=
'oblimin',verbose=True)
```

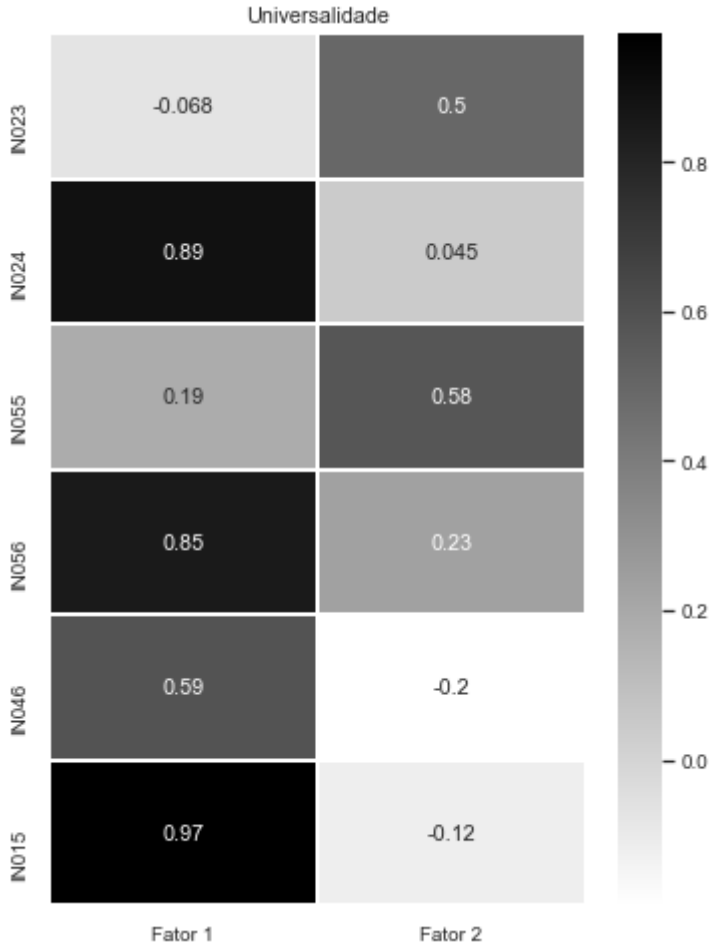

Sustentabilidade
IN002 0.0027 0.0077 -0.0055 0.0064 1 3.7e-05
Tarifa 0.0035 0.96 0.014 -0.038 -0.0095 0.11
IN101 -0.017 0.86 0.041 0.013 0.042 -0.16
IN049 -0.031 0.072 0.16 0.71 0.01 -0.17
IN057 -0.14 0.028 0.66 -0.036 0.076 0.19
IN075 0.75 0.022 -0.074 -0.04 0.036 -0.06
IN076 0.62 -0.057 0.032 0.0089 -0.0039 -0.045
IN084 0.89 0.0073 0.029 0.027 -0.017 0.062
qtde_n_micromedida -0.034 -0.003 -0.35 0.0061 0.069 -0.11
IN009 0.011 0.072 0.86 0.0042 -0.033 -0.11
IN013 0.031 -0.076 -0.1 0.86 0.02 0.13
IN029 0.021 -0.029 -0.033 0.061 0.0093 0.79
IN058 0.095 -0.32 0.35 -0.15 0.11 -0.036
Fator 1 Fator 2 Fator 3 Fator 4 Fator 5 Fator 6
0.8
0.6
0.4
0.2
0.0
−0.2

Fatores sem inverter:

	0	1	2	3	4	5
IN002	0.134722	0.174757	-0.026791	0.997800	0.082749	0.058726
Tarifa	0.934027	-0.226250	-0.312753	0.118499	0.523350	-0.234453
IN101	0.945353	-0.212151	-0.321604	0.153057	0.601201	-0.463346
IN049	0.081078	0.623457	-0.055704	0.145327	0.152154	-0.101954
IN057	0.403270	-0.123372	-0.401967	0.146001	0.667444	-0.085893
IN075	-0.232136	0.036197	0.766930	0.000022	-0.316711	-0.009328
IN076	-0.229589	0.066435	0.627636	-0.028338	-0.224242	-0.005415
IN084	-0.296303	0.119123	0.882165	-0.032327	-0.339642	0.097119
qtde_n_micromedida	-0.143972	0.055105	0.092629	0.034624	-0.286620	0.030940
IN009	0.592376	-0.201358	-0.348404	0.045334	0.937992	-0.467861
IN013	-0.375530	0.928631	0.186690	0.153011	-0.370290	0.396826
IN029	-0.337637	0.255893	0.087875	0.058132	-0.378074	0.831780
IN058	-0.095492	-0.124723	0.046968	0.068257	0.177224	-0.084134

Fatores invertidos:

	0	1	2	3	4	5
IN002	0.134722	0.174757	-0.026791	0.997800	0.082749	0.058726
Tarifa	-0.934027	0.226250	0.312753	-0.118499	-0.523350	0.234453
IN101	0.945353	-0.212151	-0.321604	0.153057	0.601201	-0.463346
IN049	-0.081078	-0.623457	0.055704	-0.145327	-0.152154	0.101954
IN057	0.403270	-0.123372	-0.401967	0.146001	0.667444	-0.085893
IN075	0.232136	-0.036197	-0.766930	-0.000022	0.316711	0.009328
IN076	0.229589	-0.066435	-0.627636	0.028338	0.224242	0.005415
IN084	0.296303	-0.119123	-0.882165	0.032327	0.339642	-0.097119
qtde_n_micromedida	-0.143972	0.055105	0.092629	0.034624	-0.286620	0.030940
IN009	0.592376	-0.201358	-0.348404	0.045334	0.937992	-0.467861
IN013	0.375530	-0.928631	-0.186690	-0.153011	0.370290	-0.396826
IN029	0.337637	-0.255893	-0.087875	-0.058132	0.378074	-0.831780
IN058	0.095492	0.124723	-0.046968	-0.068257	-0.177224	0.084134


Universalidade
IN023
IN024
IN055
IN056
IN046
IN015
-0.068
0.5
0.89
0.045
0.19
0.58
0.85
0.23
0.59
-0.2
0.97
-0.12
Fator 1
Fator 2
0.8
0.6
0.4
0.2
0.0

```
Fatores sem inverter:
             0        1
IN023 0.135873 0.472217
IN024 0.911749 0.409485
IN055 0.423678 0.655192
IN056 0.947366 0.581802
IN046 0.512650 0.046692
IN015 0.924018 0.280156

Fatores invertidos:
             0        1
IN023 0.135873 0.472217
IN024 0.911749 0.409485
IN055 0.423678 0.655192
IN056 0.947366 0.581802
IN046 0.512650 0.046692
IN015 0.924018 0.280156
------------------------------------------------- Scores ----------------
---------------------------------
     Sustentabilidade  Universalidade  Score_Medio
0            0.455424        0.246110     0.351000
1            0.361877        0.232758     0.297000
2            0.205562        0.824633     0.515000
3            0.009277        0.854862     0.432000
4            0.816236        0.078472     0.447000
..                ...             ...          ...
807          0.453120        0.332017     0.393000
808          0.890270        0.421235     0.656000
809          0.750560        0.670208     0.710000
810          0.509834        0.720318     0.615000
811          0.406040        0.200011     0.303000

[812 rows x 3 columns]
```

Comparação com anterior

In [76]:

```
cidades = ['Rio Doce','Bom Sucesso','Uberaba','Mantena','Papagaios','Lagoa da Prata','C
armópolis de Minas','Patrocínio','Monte Carmelo',
'Machado','Itaguara','São José da Varginha','Sacramento','Japaraíba','Arantina','Caraí'
,'São Sebastião do Maranhão','Setubinha',
'São João da Ponte','Presidente Bernardes','São José do Jacuri','Guaraciaba','Luisburg
o','Serra Azul de Minas','Icaraí de Minas',
'Gonçalves','Santo Antônio do Retiro','Ladainha']

cidades1=["Uberlândia",
"Araporã",
"Divinópolis",
"Pará de Minas",
"Itabirito",
"Caeté",
"Cabeceira Grande",
"Florestal",
"Monjolos",
"Pratinha"]

cidades2=["Nova Serrana"]
munic = []

for i in cidades1:
    munic.append(df[df['Nome_Município']==i].index[0])

#munic
```

In [77]:

```
df_score=pd.read_excel("C:\\Users\\DELL\\Google Drive\\2021-2\\TCC\\Codigos\\Scores.xls
x")
#df_score_dh=pd.read_excel("C:\\Users\\DELL\\Google Drive\\2021-2\\TCC\\Codigos\\Scores
_sust_univ.xlsx")
```

In [78]:

```
score_filtred=df_score.iloc[munic]
score_filtred.insert(0, 'Ranking', range(1, 1 + len(score_filtred)))

###POR CONCEITO - TABELA SCORE FINAL E MUNICIPIO
ordenado = score_filtred.sort_values('Score_Medio', ascending=False)
ordenado.insert(0, 'Rank_media', range(1, 1 + len(ordenado)))
ordenado.sort_index(ascending=True, inplace=True)
#ordenado
```

In [79]:

```
pd.set_option('display.float_format', '{:.8f}'.format)
```

In [80]:

```
##Só pra ajudar na visualização
score_mun=df["Nome_Município"]
a=score_mun.iloc[munic]
show=ordenado

ordenado = score_filtred.sort_values('Score_Medio', ascending=False)
ordenado.insert(1, 'Rank_calculado', range(1, 1 + len(ordenado)))
ordenado.sort_index(ascending=True, inplace=True)


difs = list(score_filtred.Ranking-ordenado.Rank_calculado)
show.insert(0,"Cidade",a)
show.insert(3,"Diferença",difs)
show.sort_values('Ranking')
```

Out[80]:

	Cidade	Rank_media	Ranking	Diferença	Sustentabilidade	Universalidade	Score_Me
750	Uberlândia	3	1	-2	0.51624567	0.97766203	0.74700
35	Araporã	7	2	-5	0.00004501	0.98196059	0.49100
229	Divinópolis	2	3	1	0.59074733	0.94899954	0.77000
501	Pará de Minas	4	4	0	0.43260916	0.97910825	0.70600
333	Itabirito	5	5	0	0.12337468	0.95661305	0.54000
100	Caeté	1	6	5	0.98177183	0.96712158	0.97400
94	Cabeceira Grande	9	7	-2	0.35920101	0.29699381	0.32800
268	Florestal	8	8	0	0.49206119	0.22098069	0.35700
446	Monjolos	10	9	-1	0.16372836	0.12282424	0.14300
565	Pratinha	6	10	4	0.45209380	0.55481291	0.50300

Análise Descritiva

In [81]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
from matplotlib import pyplot
from numpy import median
import numpy as np
```

In [82]:

```
## Adicionando scores no dataset
df['Score']=df_score["Score_Medio"]
df['Score_Universalidade']=df_score["Universalidade"]
df['Score_Sustentabilidade']=df_score["Sustentabilidade"]
#print(df.columns.values)
# #df
```

Melhores e piores colocados

In [83]:

```
df_aux= df.loc[:, ['Nome_Município','Score_Universalidade','Score_Sustentabilidade','Score','Natureza jurídica']]
df_aux.nlargest(10,'Score')
```

Out[83]:

	Nome_Município	Score_Universalidade	Score_Sustentabilidade	Score	Naturez jurídic
742	Três Pontas	0.98419182	0.99380505	0.98900000	Autarqui
499	Papagaios	0.98037177	0.98469913	0.98300000	Administraçã pública diret
100	Caeté	0.96712158	0.98177183	0.97400000	Autarqui
503	Paraisópolis	0.98374674	0.94982138	0.96700000	Autarqui
98	Cachoeira Dourada	0.96018107	0.96571919	0.96300000	Administraçã pública diret
502	Paraguaçu	0.96028963	0.91134938	0.93600000	Empres privad
511	Patos de Minas	0.93610403	0.92932503	0.93300000	Sociedade d economi mista cor administraçã .
58	Belo Horizonte	0.96185587	0.89543963	0.92900000	Sociedade d economi mista cor administraçã .
76	Bom Sucesso	0.93137467	0.92589218	0.92900000	Empres privad
328	Ipanema	0.89267349	0.96106665	0.92700000	Autarqui

In [84]:

```
pd.set_option('display.float_format', lambda x: '%.6f' % x)
df_aux.nsmallest (10,'Score')
```

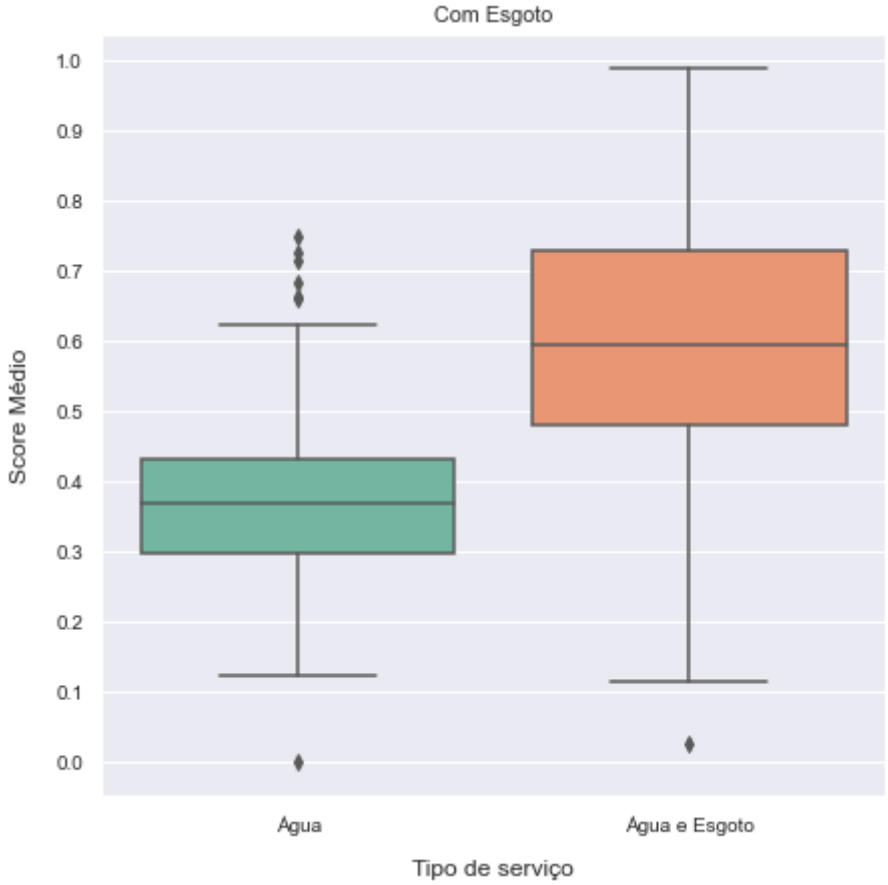
Out[84]:

	Nome_Município	Score_Universalidade	Score_Sustentabilidade	Score	Natureza jurídica
33	Arantina	0.002799	0.000000	0.001000	Sociedade de economia mista com administração ...
667	São João do Manteninha	0.049265	0.007006	0.028000	Empresa pública
182	Confins	0.230172	0.000605	0.115000	Sociedade de economia mista com administração ...
220	Descoberto	0.246321	0.001179	0.124000	Sociedade de economia mista com administração ...
223	Diogo de Vasconcelos	0.104537	0.154490	0.130000	Administração pública direta
446	Monjolos	0.122824	0.163728	0.143000	Sociedade de economia mista com administração ...
327	Ipaba	0.131929	0.214979	0.173000	Sociedade de economia mista com administração ...
669	São João do Pacuí	0.361956	0.015610	0.189000	Administração pública direta
664	São João da Ponte	0.070544	0.313555	0.192000	Sociedade de economia mista com administração ...
496	Paiva	0.382365	0.008928	0.196000	Administração pública direta

Boxplot Tipo de serviço

In [85]:

```
df_aux=df.loc[:,["Tipo de serviço",'Score']]
fig, b2 = plt.subplots(1,1, figsize=(7,7))
b2 = sns.boxplot(x="Tipo de serviço", y="Score", palette='Set2', data=df_aux)

b2.set_xlabel("Tipo de serviço", fontsize=12, labelpad=12)
b2.set_ylabel("Score Médio", fontsize=12, labelpad=12)
b2.set_yticks(np.arange(0.0, 1.1, 0.1))

b2.set_title("Com Esgoto")
plt.show()
```



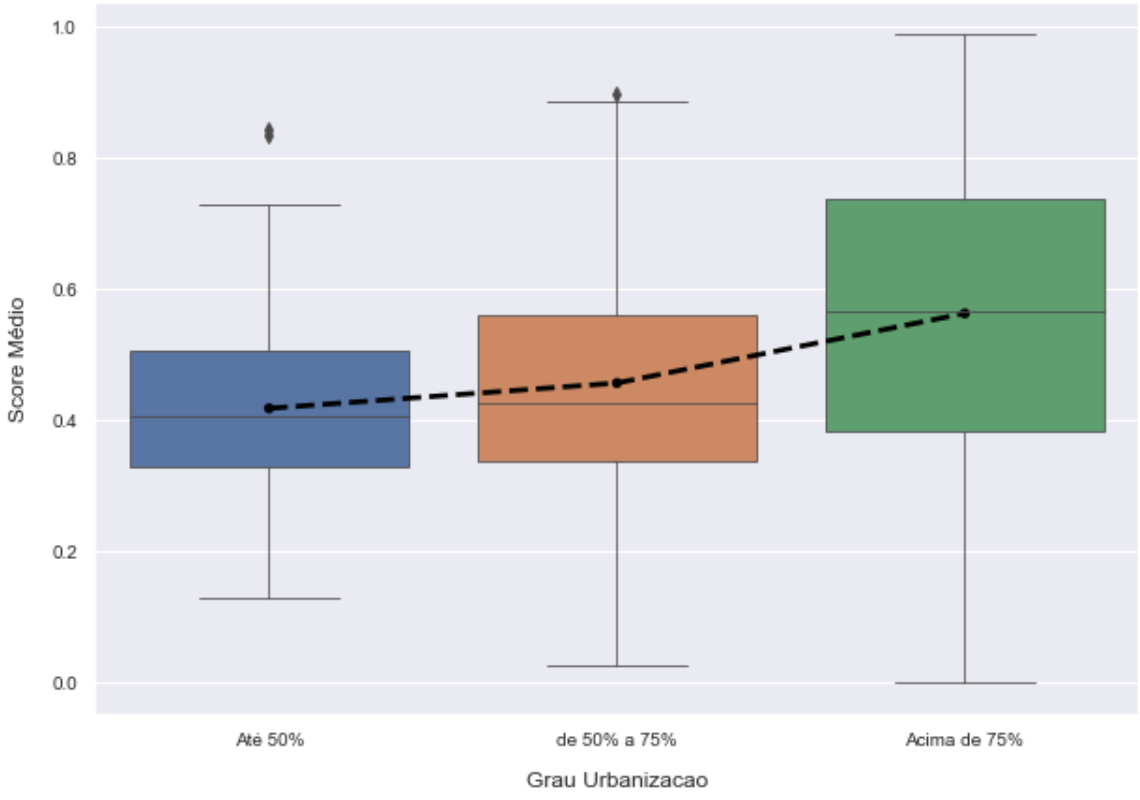
Boxplot Grau de urbanização

In [86]:

```
df_aux=df.loc[:, ['Score','grau_urbanizacao','Nome_Município']]
df_aux['Niveis_GU'] = np.where(df_aux.grau_urbanizacao<=0.50, "Até 50%",
                               np.where((df_aux.grau_urbanizacao>0.50)&(df_aux.grau_ur
banizacao<=0.75), "de 50% a 75%",
                                        "Acima de 75%") )
fig, ax = plt.subplots(1,1, figsize=(10, 7))

order=["Até 50%","de 50% a 75%","Acima de 75%"]

sns.set_style("ticks")
ax = sns.boxplot(x='Niveis_GU', y='Score', order=order, showfliers=True, linewidth=0.8,
showmeans=False, data=df_aux)
ax = sns.pointplot(x='Niveis_GU', y='Score', order=order, data=df_aux, ci=None, color=
'black', linestyles='--',markers = '.')

ax.set_xlabel("Grau Urbanizacao", fontsize=12, labelpad=12)
ax.set_ylabel("Score Médio", fontsize=12, labelpad=12)
```

Out[86]:

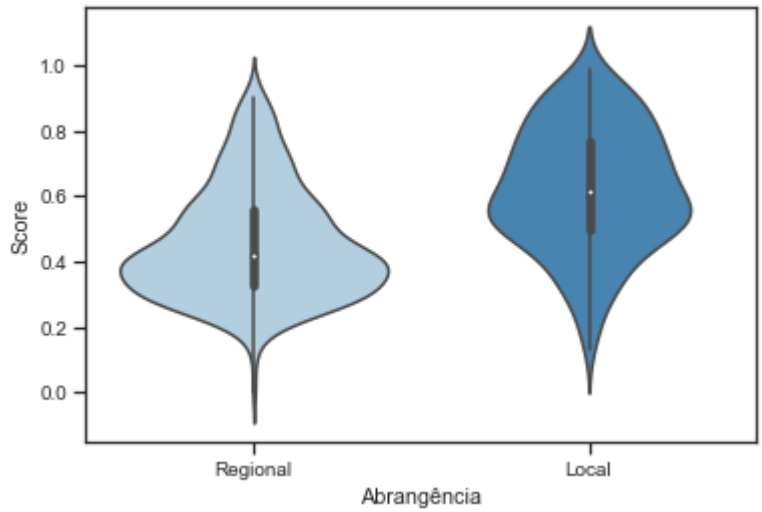
```
Text(0, 0.5, 'Score Médio')
```



Boxplot Abrangência

In [87]:

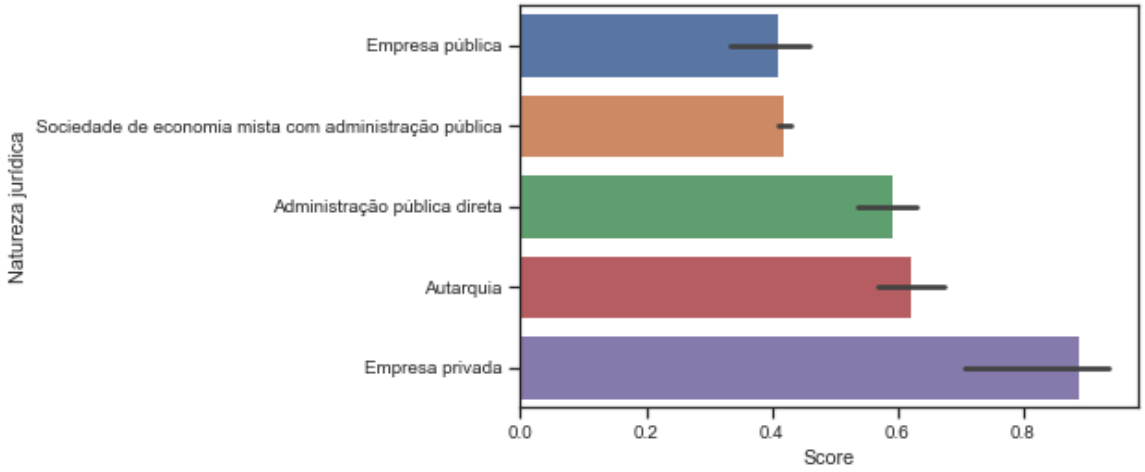
```
df_aux=df.loc[:, ['Score','Abrangência']]
violins = sns.violinplot(x="Abrangência", y="Score", data=df_aux,widths=0.45, palette=
'Blues')
```



Natureza juridica (Mediana)

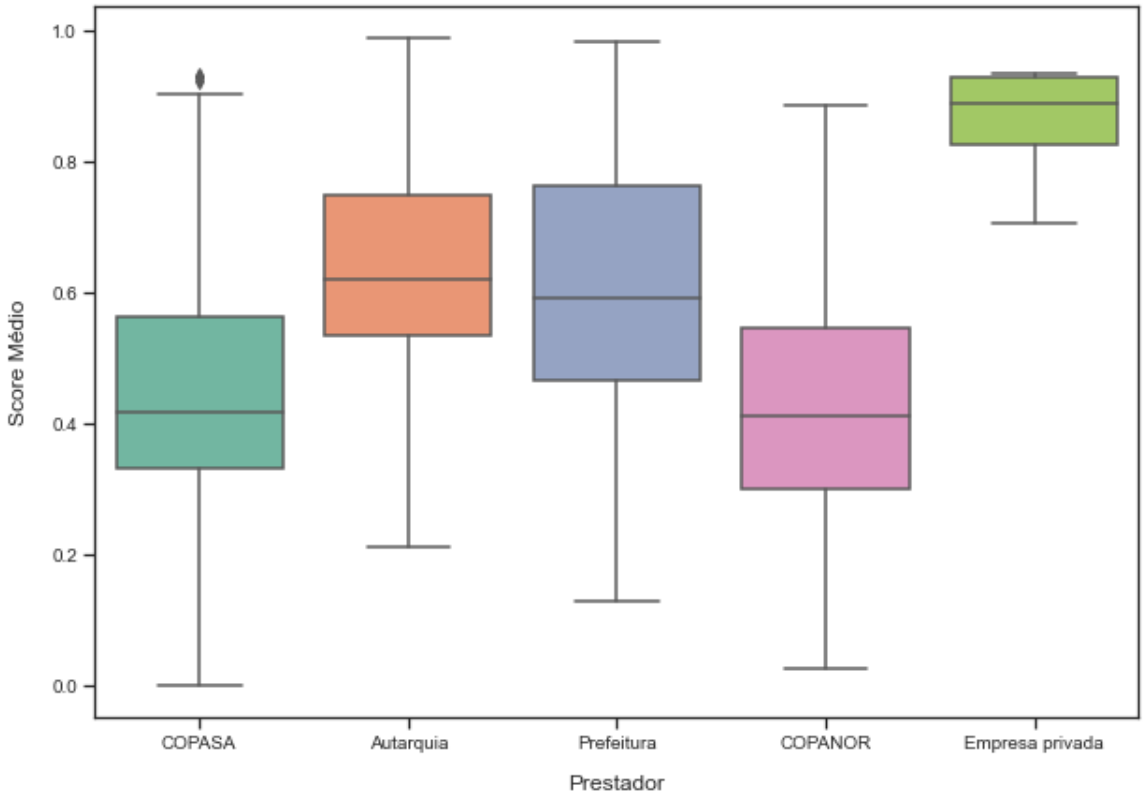
In [88]:

```
df_aux=df.loc[:, ['Score','Natureza jurídica']]
order=df_aux.groupby(["Natureza jurídica"])["Score"].median().sort_values().index
p=sns.barplot(x='Score', y='Natureza jurídica', data=df_aux, estimator=np.median, order
=order)
```



In [89]:

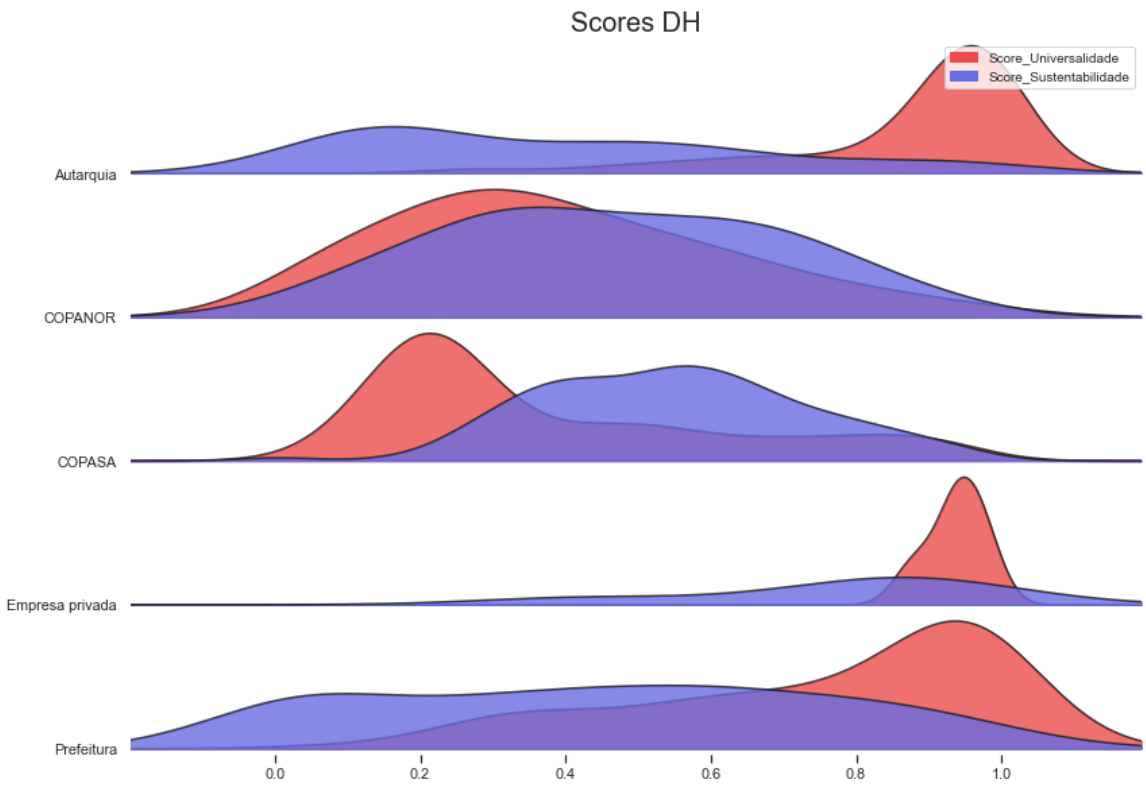
```
df_aux=df.loc[:, ['Score','Prestador2']]

fig, b2 = plt.subplots(1,1, figsize=(10, 7))
b2 = sns.boxplot(x="Prestador2", y="Score", palette='Set2', data=df_aux)

b2.set_xlabel("Prestador", fontsize=12, labelpad=12)
b2.set_ylabel("Score Médio", fontsize=12, labelpad=12)

plt.show()
```



In [90]:

```
from joypy import joyplot

df_aux=df.loc[:, ['Score_Universalidade','Score_Sustentabilidade','Prestador2']]
# df_aux=df_aux.melt(id_vars=["prestador2"],
#         var_name="Tipo_Score",
#         value_name="Scores")
# df_aux["Merge"]=df_aux["prestador2"]+" "+df_aux["Tipo_Score"]

plt.figure()

ax, fig = joyplot(
    data=df_aux,
    by='Prestador2',
    column=['Score_Universalidade','Score_Sustentabilidade'],
    color=[ '#eb4d4b','#686de0',"#f37b2d"],
    legend=True,
    alpha=0.80,
    figsize=(12, 8),
    ylim='own',
    overlap=0
)
plt.title('Scores DH', fontsize=20)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

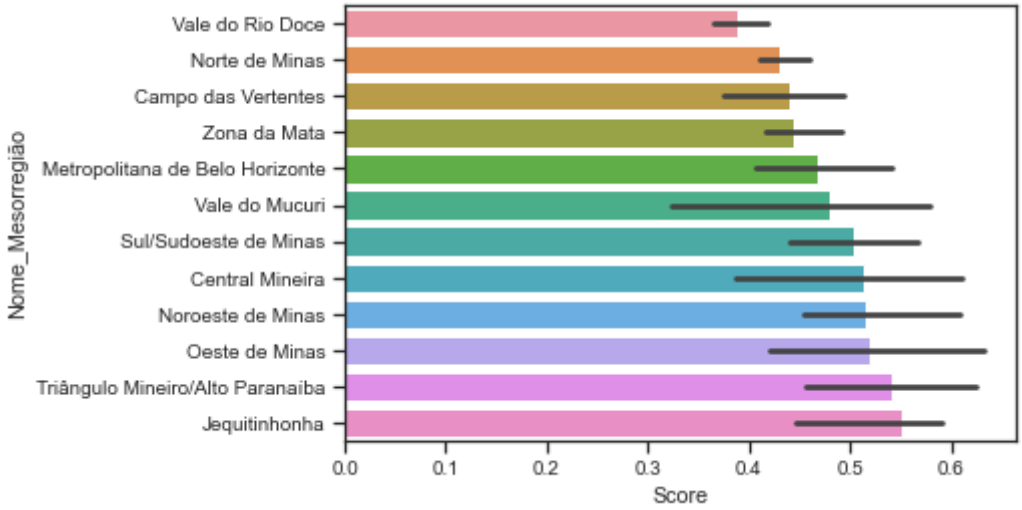



```
<Figure size 432x288 with 0 Axes>
```

Mesoregião (Mediana)

In [91]:

```
df_aux=df.loc[:, ['Score','Nome_Mesorregião']]
order=df_aux.groupby(["Nome_Mesorregião"])["Score"].median().sort_values().index
p=sns.barplot(x='Score', y='Nome_Mesorregião', data=df_aux, estimator=np.median, order=
order)
```



In [92]:

```
df_aux=df.loc[:, ['Score','Nome_Mesorregião']]
#df_aux.groupby('NV08Meso')['Score'].median()
df_aux=df_aux.groupby(df_aux.Nome_Mesorregião)[['Score']].median()
df_aux.sort_values('Score')
```

Out[92]:

Nome_Mesorregião	Score
Vale do Rio Doce	0.388000
Norte de Minas	0.431000
Campo das Vertentes	0.441000
Zona da Mata	0.444500
Metropolitana de Belo Horizonte	0.467500
Vale do Mucuri	0.480000
Sul/Sudoeste de Minas	0.503500
Central Mineira	0.514500
Noroeste de Minas	0.516500
Oeste de Minas	0.520500
Triângulo Mineiro/Alto Paranaíba	0.542000
Jequitinhonha	0.551500

In [93]:

```
import shapefile as shp
from unidecode import unidecode
import geopandas as gpd
import folium
from folium import plugins
import json
import branca.colormap as cmp
import shapely.geometry
import plotly.express as px
```

In [94]:

```
with open("C:/Users/DELL/Google Drive/2021-2/TCC/geojs-31-mun.json", encoding="utf8") a
s file:
    geo_json_data = json.load(file)
```

In [95]:

```
df_media = pd.DataFrame( {'name': df['Nome_Município'], 'Score':df['Score'],'Score_Univ
ersalidade':df['Score_Universalidade'],
                          'Score_Sustentabilidade':df['Score_Sustentabilidade']})
df_MG = df_media[~df_media.duplicated(subset=['name'], keep='first')]
#df_MG.loc[:, 'Cidade'] = df_MG.loc[:, 'Cidade'].str.upper()

df_MG['Quantil'] = np.where(df_MG.Score<=0.25, "(0-25)%",
                            np.where((df_MG.Score>0.25)&(df_MG.Score<=0.5), "(25-50)%",
                                     np.where((df_MG.Score>0.5)&(df_MG.Score<=0.75), "
(50-75)%", "(75-100)%")
                                    )
                           )
with open("C:/Users/DELL/Google Drive/2021-2/TCC/geojs-31-mun.json", encoding="utf8") a
s file:
    geo_json_data = json.load(file)

geo_df = gpd.GeoDataFrame.from_features(geo_json_data["features"]).merge(df_MG, on="nam
e").set_index("name")
#geo_df
```

In [96]:

```
# import plotly.io as pio
# pio.renderers.default = 'browser'
```

Mapa Score

In [97]:

```
df_media = pd.DataFrame( {'name': df['Nome_Município'], 'Score':round(df['Score'],4),
                          'Score_Universalidade':round(df['Score_Universalidade'],4),
                          'Score_Sustentabilidade':round(df['Score_Sustentabilidade'],4
),
                          'Nome_Mesorregião':df['Nome_Mesorregião']})
df_MG = df_media[~df_media.duplicated(subset=['name'], keep='first')]
#df_MG.loc[:, 'Cidade'] = df_MG.loc[:, 'Cidade'].str.upper()

df_MG['Quantil'] = np.where(df_MG.Score<=0.25, "(0-25)%",
                            np.where((df_MG.Score>0.25)&(df_MG.Score<=0.5), "(25-50)%",
                                     np.where((df_MG.Score>0.5)&(df_MG.Score<=0.75), "
(50-75)%", "(75-100)%")
                                    )
                           )
with open("C:/Users/DELL/Google Drive/2021-2/TCC/geojs-31-mun.json", encoding="utf8") a
s file:
    geo_json_data = json.load(file)

geo_df = gpd.GeoDataFrame.from_features(geo_json_data["features"]).merge(df_MG, on="nam
e").set_index("name")
fig = px.choropleth_mapbox(geo_df,
                           geojson=geo_df.geometry,
                           locations=geo_df.index,
                           color="Quantil",
                           category_orders= {'Quantil':["(0-25)%","(25-50)%","(50-75)%"
, "(75-100)%"]},
                           color_discrete_sequence=["#922B21", "#E67E22", "#F4D03F"," #
52BE80"],
                           center={"lat": -19.84164, "lon": -43.98651},
                           mapbox_style="open-street-map",
                           zoom=6,
                           #hover_name='name',
                           hover_data=['Quantil','Score','Score_Universalidade','Score_
Sustentabilidade'],
                           title="Score"
                          )
fig.show()
```

Mapa Universalidade

In [98]:

```
df_media = pd.DataFrame( {'name': df['Nome_Município'], 'Score':round(df['Score'],4),
                          'Score_Universalidade':round(df['Score_Universalidade'],4),
                          'Score_Sustentabilidade':round(df['Score_Sustentabilidade'],4
),
                          'Nome_Mesorregião':df['Nome_Mesorregião']})
df_MG = df_media[~df_media.duplicated(subset=['name'], keep='first')]
#df_MG.loc[:, 'Cidade'] = df_MG.loc[:, 'Cidade'].str.upper()

df_MG['Quantil'] = np.where(df_MG.Score_Universalidade<=0.25, "(0-25)%",
                            np.where((df_MG.Score_Universalidade>0.25)&(df_MG.Score_Uni
versalidade<=0.5), "(25-50)%",
                                     np.where((df_MG.Score_Universalidade>0.5)&(df_MG.S
core_Universalidade<=0.75), "(50-75)%", "(75-100)%")
                                    )
                           )
with open("C:/Users/DELL/Google Drive/2021-2/TCC/geojs-31-mun.json", encoding="utf8") a
s file:
    geo_json_data = json.load(file)

geo_df = gpd.GeoDataFrame.from_features(geo_json_data["features"]).merge(df_MG, on="nam
e").set_index("name")
fig = px.choropleth_mapbox(geo_df,
                           geojson=geo_df.geometry,
                           locations=geo_df.index,
                           color="Quantil",
                           category_orders= {'Quantil':["(0-25)%","(25-50)%","(50-75)%"
, "(75-100)%"]},
                           color_discrete_sequence=["#922B21", "#E67E22", "#F4D03F"," #
52BE80"],
                           center={"lat": -19.84164, "lon": -43.98651},
                           mapbox_style="open-street-map",
                           zoom=6,
                           #hover_name='name',
                           hover_data=['Quantil','Score','Score_Universalidade','Score_
Sustentabilidade'],
                           title="Score_Universalidade"
                          )
fig.show()
```

Mapa Sustentabilidade

In [99]:

```
df_media = pd.DataFrame( {'name': df['Nome_Município'], 'Score':round(df['Score'],4),
                          'Score_Universalidade':round(df['Score_Universalidade'],4),
                          'Score_Sustentabilidade':round(df['Score_Sustentabilidade'],4
),
                          'Nome_Mesorregião':df['Nome_Mesorregião']})
df_MG = df_media[~df_media.duplicated(subset=['name'], keep='first')]
#df_MG.loc[:, 'Cidade'] = df_MG.loc[:, 'Cidade'].str.upper()

df_MG['Quantil'] = np.where(df_MG.Score_Sustentabilidade<=0.25, "(0-25)%",
                            np.where((df_MG.Score_Sustentabilidade>0.25)&(df_MG.Score_S
ustentabilidade<=0.5), "(25-50)%",
                                     np.where((df_MG.Score_Sustentabilidade>0.5)&(df_MG
.Score_Sustentabilidade<=0.75), "(50-75)%", "(75-100)%")
                                    )
                           )
with open("C:/Users/DELL/Google Drive/2021-2/TCC/geojs-31-mun.json", encoding="utf8") a
s file:
    geo_json_data = json.load(file)

geo_df = gpd.GeoDataFrame.from_features(geo_json_data["features"]).merge(df_MG, on="nam
e").set_index("name")
fig = px.choropleth_mapbox(geo_df,
                           geojson=geo_df.geometry,
                           locations=geo_df.index,
                           color="Quantil",
                           category_orders= {'Quantil':["(0-25)%","(25-50)%","(50-75)%"
, "(75-100)%"]},
                           color_discrete_sequence=["#922B21", "#E67E22", "#F4D03F"," #
52BE80"],
                           center={"lat": -19.84164, "lon": -43.98651},
                           mapbox_style="open-street-map",
                           zoom=6,
                           #hover_name='name',
                           hover_data=['Quantil','Score','Score_Universalidade','Score_
Sustentabilidade'],
                           title="Score_Sustentabilidade"
                          )
fig.show()
```