AnalisisyCuracion_Parte-III_Description

August 13, 2020

0.0.1 DESCRIPTORES DE MERCADO POR SEGMENTO

ANALISIS DE TRANSACCIONES E INFRAESTRUCTURAS POR SEGMENTO DE MERCADO

By Diego Tondo, August 2020

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
np.set_printoptions(precision=2)
pd.set_option("display.precision", 2)
```

[ ]:
```
dft = pd.read_csv('../dataset/data_csv/sis_transa_201801_202007_merged.csv.zip',
                  parse_dates=['fecha'],
                  compression='zip')
dft.shape
```

[ ]:
```
dft['id_empresa'] = dft['id_empresa'].astype('int')
dft = dft[(dft['segmento']!='d')]
dft.shape
```

[ ]:
```
dft.columns
```

[ ]:
```
dft18 = dft[ (dft['fecha']<'2019-01-01') ]
dft19 = dft[ (dft['fecha']<'2020-01-01') & (dft['fecha']>='2019-01-01') ]
dft20 = dft[ (dft['fecha']>='2020-01-01') ]
```

[ ]:
```
dft18.shape, dft19.shape, dft20.shape
```

DESCRIPTORES GENERALES DE MERCADO

[ ]:
```
## CANTIDAD DE EMPRESAS POR AÑO
qemp = np.asarray([dft18['id_empresa'].nunique(), dft19['id_empresa'].
 ↪nunique(), dft20['id_empresa'].nunique()])
print(qemp)
```

[ ]:
```
## CANTIDAD DE EMPRESAS POR AÑO
scant = np.asarray([dft18['cantidad'].sum(), dft19['cantidad'].sum(),␣
 ↪dft20['cantidad'].sum()])
print(f'Litros controlados por año: {scant}')
```

[ ]:
```
print(f'Litros controlados en 20 respecto de 19: %{scant[2]/scant[1]*100}')
```

[ ]:
```
print(f'Litros controlados por empresa: {scant/qemp}')
```

[ ]:
```
print(f'Litros controlados por empresa por mes: {scant/qemp/12}')
```

[ ]:
```
## CANTIDAD DE SITIOS POR EMPRESA
qsites = np.asarray([dft18['id_equipo'].nunique(), dft19['id_equipo'].
 ↪nunique(), dft20['id_equipo'].nunique()])
qsites
```

[ ]:
```
print(f'Sitios controlados por empresa: {qsites/qemp}')
```

[ ]:
```
qtanks = np.asarray([dft18['id_tanque'].nunique(), dft19['id_tanque'].
 ↪nunique(), dft20['id_tanque'].nunique()])
qtanks
```

[ ]:
```
print(f'Tanques controlados por empresa: {qtanks/qemp}')
```

[ ]:
```
print(f'Tanques controlados por sitio: {qtanks/qsites}')
```

DESCRIPTORES DE TRANSACCIONES POR SEGMENTO DE MERCADO

[ ]:
```
def TransactionsGroupedBySegment(df):
    dfg = df.groupby(['segmento'])['cantidad'].describe()
    dfg['count %'] = dfg['count']/dfg['count'].sum()*100
    dfg = dfg.transpose()
    return dfg
```

[ ]:
```
for i in (18, 19, 20):
    vars()[f'dft{i}_s_qtran'] = TransactionsGroupedBySegment(eval(f'dft{i}'))
```

[ ]:
```
dft19_s_qtran
```

[ ]:
```
dft18_s_qtran
```

[ ]:
```
dft19_s_qtran
```

[ ]:
```
dft20_s_qtran
```

[ ]:
```
## SAVE DATA TO CSV
#[eval(f'dft{i}_s_qtran').to_csv(f'../results/dft{i}_s_qtran.csv') for i in␣
 ↪(18, 19, 20)]
```

BOXPLOT

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

[ ]:
```
sns.set_style("whitegrid", {"grid.color": ".9"})
```

[ ]:
```
segList = ['Agriculture', 'Construction', 'Service Stations', 'Industry',␣
 ↪'Mining',  'Oil&Gas', 'Telcos', 'Transportation']
```

[ ]:
```
dft19 = dft19.sort_values(by=['segmento'])
```

[ ]:
```
fig = plt.figure(figsize=(16,8))
```

```
ax = sns.boxplot(data=dft19, x='segmento', y='cantidad', showmeans=True,␣
 ↪meanline=True,
                 flierprops = dict(markerfacecolor='0.5', markersize=0.1,␣
 ↪linestyle='none'))

plt.title("Volumen por Transacción, por Segmento",fontsize=16)
plt.ylabel('Litros por Transaccion [Lts]',fontsize=16)
plt.xlabel('Segmento', fontsize=14)

major_ticks = np.arange(0, 900, 50)
minor_ticks = np.arange(0, 900, 10)

plt.grid(True)
ax.set_yticks(major_ticks)
ax.set_yticks(minor_ticks, minor=True)
#plt.xticks(fontsize=16)
plt.xticks(np.arange(8), segList, fontsize=14, rotation=0)
plt.yticks(fontsize=14)

plt.ylim(0, 900)

ax.grid(which='both')
ax.grid(which='minor', alpha=0.2)
ax.grid(which='major', alpha=0.75)

#sns.despine()
plt.show()
```

[ ]:
```
#fig.savefig('../results/boxplot_volxtransaxseg.png', bbox_inches = 'tight')
```

CONSUMO POR EMPRESA

[ ]:
```
def TransactionsGroupedBySegAndCompany(dft):
    dft_cons = dft.groupby(['segmento','id_empresa'])['cantidad'].sum()
    ctot = dft_cons.sum()
    dft_s_vtran = dft_cons.groupby('segmento').describe()
    etot = dft19['id_empresa'].nunique()
    dft_s_vtran['count %'] = dft_s_vtran['count']/etot*100
    dft_s_vtran['mean monthly'] = dft_s_vtran['mean']/12
    dft_s_vtran['vol'] = dft_cons.groupby('segmento').sum()
    dft_s_vtran['vol %'] = dft_s_vtran['vol']/ctot*100
    dft_s_vtran['vol per comp'] = dft_s_vtran['vol']/dft_s_vtran['count']
    dft_s_vtran['vol monthly'] = dft_s_vtran['vol']/12
    dft_s_vtran = dft_s_vtran.transpose()
    return (dft_s_vtran, dft_cons)
```

[ ]:
```
for i in 18, 19, 20:
```

```
    vars()[f'dft{i}_s_vtran'], vars()[f'dft{i}_cons'] =␣
↪TransactionsGroupedBySegAndCompany(eval(f'dft{i}'))
```

[ ]:
```
dft18_s_vtran
```

[ ]:
```
dft19_s_vtran
```

[ ]:
```
dft20_s_vtran
```

[ ]:
```
## SAVE DATA TO CSV
[eval(f'dft{i}_s_vtran').to_csv(f'../results/dft{i}_s_vtran.csv') for i in (18,␣
↪19, 20)]
```

BOXPLOT

[ ]:
```
dfc = dft19_cons.reset_index()
```

[ ]:
```
dfc['cantidad m'] = dfc['cantidad']/12
```

[ ]:
```
fig = plt.figure(figsize=(16,8))

ax = sns.boxplot(data=dfc, x='segmento', y='cantidad m', showmeans=True,␣
↪meanline=True,
                 flierprops = dict(markerfacecolor='0.5', markersize=1,␣
↪linestyle='none'))

plt.title("Volumen Mensual Transaccionado por Empresa por Segmento",fontsize=16)
plt.ylabel('Litros Transaccionados [Lts]',fontsize=16)
plt.xlabel('Segmento', fontsize=14)

major_ticks = np.arange(0, 350000, 25000)
minor_ticks = np.arange(0, 350000, 5000)
plt.ylim(0, 350000)

plt.grid(True)
ax.set_yticks(major_ticks)
ax.set_yticks(minor_ticks, minor=True)
plt.xticks(fontsize=16)
plt.xticks(np.arange(8), segList, fontsize=14, rotation=0)
plt.yticks(fontsize=14)

ax.grid(which='both')
ax.grid(which='minor', alpha=0.2)
ax.grid(which='major', alpha=0.75)

#sns.despine()
plt.show()
```

[ ]:
```
#fig.savefig('../results/boxplot_volxempresaxseg.png', bbox_inches = 'tight')
```

[ ]:
```

```

ANALISIS DE INFRAESTRUCTURAS POR SEGMENTO

[ ]:
```
def infra(df):
    df_infra = df.groupby(['segmento','id_empresa'])['id_equipo', 'id_tanque',␣
 ↪'main_id'].nunique()
    df_infra = df_infra.groupby('segmento').mean().transpose()
    return df_infra
```

[ ]:
```
for i in 18, 19, 20:
    vars()[f'dft{i}_infra'] = infra(eval(f'dft{i}'))
```

[ ]:
```
dft20_infra, dft19_infra, dft18_infra
```

[ ]:
```
## SAVE DATA TO CSV
#dft_infra19.to_csv('../results/dft19_infra.csv')
```

Distribuciones por Segmento

[ ]:
```
for y in (18, 19, 20):
    df = eval(f'dft{y}')
    for i in df['segmento'].unique():
        vars()[f'{i}{y}'] = df[df['segmento'].str.contains(str(i), regex=True)].
 ↪groupby('id_empresa')['cantidad'].sum()/12
        vars()[f'{i}{y}'] = vars()[f'{i}{y}'].reset_index()
        vars()[f'{i}{y}'].name = f'{i}{y}'
```

[ ]:
```
def histograma(df):
    fig = plt.figure(figsize=(16,8))

    sns.distplot(df['cantidad'], bins=30)

    plt.title(f'Histograma de Empresas tipo {df.name} por Volumen␣
 ↪Mensual',fontsize=16)
    plt.ylabel('Empresas',fontsize=16)
    plt.xlabel('Litros [Lts]', fontsize=14)
    plt.grid(axis='y', alpha=0.75)

    #plt.ylim(0, 50)
    plt.xlim(0, 200000)
    plt.xticks(fontsize=16)
    plt.yticks(fontsize=14)

    plt.show()
```

[ ]:
```
histograma(t20), histograma(t19), histograma(t18)
```

[ ]:
```
#fig.savefig('../results/hist_t19.png', bbox_inches = 'tight')
```

[ ]:
```
histograma(o20), histograma(o19), histograma(o18)
```

[ ]:
```

```