

AnalisisCuracion_Parte-IV_Tendencias

August 13, 2020

0.1 ANALISIS DE TENDENCIAS DE CONSUMOS E INFRAESTRUCTURAS

By JoseForguez & DiegoTondo, 2020-08-09

```
[ ]: import pandas as pd
import numpy as np
import warnings
import glob
import zipfile
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime

import plotly.graph_objs as go
import plotly.offline as plotly
from plotly.subplots import make_subplots

import matplotlib.dates as md

[ ]: #pd.set_option('display.max_columns', 50)
```

LOAD TRANSACTIONS

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

[ ]: df_transa.columns
```

0.2 CONSUMPTION IN TIME

```
[ ]: df_transa_pos = df_transa.copy()

[ ]: dfgroupby_equipo = pd.DataFrame()
# grupo = ['id_equipo', 'mes']
# grupo = 'id_equipo'
grupo = pd.Grouper(key="fecha", freq="M")

dfgroupby_equipo['cantidad'] = df_transa_pos.groupby(grupo)['cantidad'].sum()
dfgroupby_equipo['main_id'] = df_transa_pos.groupby(grupo)['main_id'].nunique()
```

```
dfgroupby_equipo['id_bomba'] = df_transa_pos.groupby(grupo)['id_bomba'].
    →nunique()
```

```
[ ]: dfgroupby_equipo.reset_index(inplace=True)
```

```
[ ]: # sp2.set_title("Cantidad de litros despachados")
#ax=sns.distplot(dfgroupby_equipo['cantidad'],kde=False, norm_hist=False,
    →label='volumen suministrado[Lts]')
fig = plt.figure(figsize=(15,8))

df = dfgroupby_equipo

# with sns.axes_style("whitegrid"):
sns.set_style("whitegrid", {"grid.color": ".9"})
ax1 = sns.lineplot(x=df['fecha'],y=df['cantidad']/1e6, linewidth=3,
    →color='deepskyblue')
#plt.scatter(xm,ym, s=100, c='g', alpha=0.4, edgecolors="g", linewidth=5)
#plt.scatter(xm_covid, ym_covid, s=100, c='g', alpha=0.4, edgecolors="g",
    →linewidth=5)
#plt.scatter(xmm,ymm, s=100, c='g', alpha=0.4, edgecolors="g", linewidth=5)

ym = df['cantidad'].max()/1e6
xm = df['fecha'][df['cantidad'].idxmax()]
ymm = df['cantidad'].min()/1e6
xmm = df['fecha'][df['cantidad'].idxmin()]
xm_covid = pd.to_datetime('2020-01-31')
ym_covid = df.iloc[df.set_index('fecha').index.get_loc(xm_covid,
    →method='nearest')]['cantidad']/1e6

# Max
plt.annotate('{}\nElecciones PASO 2019\n{:.2f} Mill. de Litros'.format(xm.
    →strftime("%b %Y"), ym),
            xy=(xm,ym),
            xytext=(xm, ym*1.2),
            fontsize=14,
            ha='center',
            arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
    →width=2))

plt.annotate('{}\nCovid-19\n{:.2f} Mill. de Litros'.format(xm_covid.
    →strftime("%b %Y"), ym_covid),
            xy=(xm_covid, ym_covid),
            xytext=(xm_covid, ym_covid+2),
            fontsize=14,
            ha='center',
            arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
    →width=2))
```

```

# Min
plt.annotate('{}\n{: .2f} Mill. de Litros'.format(xmm.strftime("%b %Y"), ymm),
             xy=(xmm, ymm),
             xytext=(xmm, ymm-3),
             fontsize=14,
             ha='center',
             arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
                               width=2))
#bbox_props = dict(
#     boxstyle="arrow,pad=0.5",
#     fc="cyan", ec="b", lw=1
#)
#plt.text(xm, ym-2, "Politics & Covid Effect",
#         ha="left", va="top",
#         rotation=-35,
#         size=15,
#         bbox=bbox_props)
# ax1.annotate('local max', xy=(3, 1), xycoords='data',
#             xytext=(0.8, 0.95), textcoords='axes fraction',
#             arrowprops=dict(facecolor='black', shrink=0.05),
#             horizontalalignment='right', verticalalignment='top',
#             )
plt.ylabel('Millones de Litros', fontsize=16)
plt.xlabel('')
plt.title("Volumen mensual despachado y controlado por Cintelink", fontsize=16)
plt.ylim(0, 25)
# ax.set_yscale('log')
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))
plt.xticks(df['fecha'], fontsize=12, rotation='90')
plt.yticks(fontsize=14)
plt.grid(True)
sns.despine()
plt.show()
# fig.savefig('report_volumen.pdf')

```

```
[ ]: df.groupby_equipos
```

```
[ ]: df = df.groupby_equipos
```

```

fig = plt.figure(figsize=(15, 8))

ax1 = sns.lineplot(x=df['fecha'], y=df['main_id'], linewidth=3,
                  color='deepskyblue')

plt.ylabel('Vehiculos Activos', fontsize=16)
plt.xlabel('Fecha', fontsize=14)
plt.title("Cantidad mensual de vehículos que cargaron al menos una vez en
          Cintelink", fontsize=16)

```

```

plt.ylim(4,8)
# ax.set_yscale('log')
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))
plt.xticks(df['fecha'],fontsize=12, rotation='90')
plt.yticks(fontsize=14)
plt.grid(True)
sns.despine()

plt.show()

```

```

[: fig = plt.figure(figsize=(15,8))

df = dfgroupby_equipo[dfgroupby_equipo['fecha']>'2019-08-01']

y = df['cantidad']/df['main_id']

ax1=sns.lineplot(x=dfgroupby_equipo['fecha'],
                 y=dfgroupby_equipo['cantidad']/dfgroupby_equipo['main_id'],
                 linewidth=3, color='deepskyblue')

ym = y.max()
ymim = y.min()
xm = df['fecha'][y.idxmax()]

xm_covid = pd.to_datetime('2020-01-31')

ym_covid = y.iloc[df.set_index('fecha').index.get_loc(xm_covid,
method='nearest')]

# Max
plt.annotate('{}\nElecciones PASO 2019'.format(xm.strftime("%b %Y")),
            xy=(xm,ym),
            xytext=(xm, ym*1.1),
            fontsize=14,
            ha='center',
            arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
width=2))

plt.annotate('{}\nCovid-19'.format(xm_covid.strftime("%b %Y")),
            xy=(xm_covid, ym_covid),
            xytext=(xm_covid, ym_covid*1.1),
            fontsize=14,
            ha='center',
            arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
width=2))

```

```

# Min
# plt.annotate('{ }\n{:.2f} Mill. de Litros'.format(xmm.strftime("%b %Y"), ymm),
#             xy=(xmm, ymm),
#             xytext=(xmm, ymm-3),
#             fontsize=14,
#             ha='center',
#             arrowprops=dict(facecolor='black', shrink=0.0, headwidth=3,
#                               →width=2))

plt.ylabel('Litros por mes', fontsize=16)
plt.xlabel('Fecha', fontsize=14)
plt.title("Consumo promedio mensual por vehículo controlado por
→Cintelink", fontsize=16)
plt.ylim(ymim*0.9, ym*1.2)
# ax.set_yscale('log')
plt.xticks(dfgroupby_equipo['fecha'], fontsize=12, rotation='90')
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))
plt.yticks(fontsize=14)
plt.grid(True)
sns.despine()

plt.show()

```

```

[: fig = plt.figure(figsize=(15,8))

df = dfgroupby_equipo
ax1=sns.lineplot(x=df['fecha'], y=df['id_bomba'], linewidth=3,
→color='deepskyblue')

plt.ylabel('Cantidad de bombas activas', fontsize=16)
plt.xlabel('Fecha', fontsize=14)
plt.title("Cantidad mensual de bombas activas controladas por
→Cintelink", fontsize=16)
# plt.ylim(4,8)
# ax.set_yscale('log')
plt.xticks(df['fecha'], fontsize=12, rotation='90')
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))
plt.yticks(fontsize=12)
plt.grid(True)
sns.despine()

plt.show()

```

```
[ ]:
```

0.3 ANALISIS DE HISTORIAL DE TANQUES

```
[ ]: df_historial = pd.read_csv('../dataset/data_csv/historial_tanques_ultra_light.  
    ↳zip', parse_dates=['fecha_hora'], compression='zip')  
df_historial.info()
```

```
[ ]: dfgroupby_historial = pd.DataFrame()  
# grupo = ['id_equipo', 'mes']  
# grupo = 'id_equipo'  
grupo = pd.Grouper(key="fecha_hora", freq="M")  
  
dfgroupby_historial['id_tanque'] = df_historial.groupby(grupo)['id_tanque'].  
    ↳nunique()  
dfgroupby_historial['volumen'] = df_historial.groupby(grupo)['volumen'].mean()  
dfgroupby_historial.reset_index(inplace=True)
```

```
[ ]: fig = plt.figure(figsize=(15,8))  
  
ax1=sns.  
    ↳lineplot(x=dfgroupby_historial['fecha_hora'],y=dfgroupby_historial['id_tanque'],  
    ↳linewidth=3, color='Red')  
  
plt.ylabel('Cantidad de Tanques activos',fontsize=16)  
plt.xlabel('Fecha',fontsize=14)  
plt.title("Cantidad mensual de tanques activos controlados por_  
    ↳Cintelink",fontsize=16)  
plt.ylim(50,400)  
# ax.set_yscale('log')  
plt.xticks(dfgroupby_historial['fecha_hora'],fontsize=12, rotation='90')  
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))  
plt.yticks(fontsize=12)  
plt.grid(True)  
sns.despine()  
  
plt.show()
```

```
[ ]: fs_tanque = pd.read_csv('../dataset/data_csv/fs_tanques.csv')  
fs_tanque.info()
```

```
[ ]: dfgroupby_tanques = pd.DataFrame()  
# grupo = ['id_equipo', 'mes']  
# grupo = 'id_equipo'  
grupo = [pd.Grouper(key="fecha_hora", freq="M"), 'id_tanque']
```

```

# dfgroupby_tanques['id_tanque'] = df_historiales.groupby(grupo)['id_tanque'].
  ↳nunique()
dfgroupby_tanques['volumen'] = df_historial.groupby(grupo)['volumen'].mean()

[: index=fs_tanque.set_index('id_tanque').index.get_indexer(dfgroupby_tanques.
  ↳reset_index('fecha_hora').index)

[: dfgroupby_tanques['capacidad']=fs_tanque.iloc[index]['capacidad'].values

[: dfgroupby_tanques['capacidad']=dfgroupby_tanques['capacidad'].astype('float')

[: grupo = pd.Grouper(key="fecha_hora", freq="M")
dfgroupby_capacidades=pd.DataFrame()
dfgroupby_capacidades['capacidad']=dfgroupby_tanques.reset_index().
  ↳groupby(grupo)['capacidad'].sum()
dfgroupby_capacidades.reset_index(inplace=True)

[: fig = plt.figure(figsize=(15,8))

ax1=sns.
  ↳lineplot(x=dfgroupby_capacidades['fecha_hora'],y=dfgroupby_capacidades['capacidad']/
  ↳1e6, linewidth=3, color='Red')

plt.ylabel('Millones de Litros',fontsize=16)
plt.xlabel('',fontsize=14)
plt.title("Volumen Mensual controlado por Sondas de Telemedición_
  ↳Cintelink",fontsize=16)
# plt.ylim(50,400)
# ax.set_yscale('log')
plt.xticks(dfgroupby_historial['fecha_hora'],fontsize=12, rotation='90')
ax1.xaxis.set_major_formatter(md.DateFormatter('%b %Y'))
plt.yticks(fontsize=12)
plt.grid(True)
sns.despine()

plt.show()

[:

```