memoria

February 5, 2022

1 Libraries

[ ]:
```
#        DATA
# ==================== #
import pandas as pd
import numpy as np
from scipy import stats
import time
import random
import math

#      PLOTING
# ============================== #
from IPython.display import Image
from IPython.core.display import HTML

import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="white")

#      TimeSeries
#␣
 ↪==========================================================================␣
 ↪#
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import acf , pacf
from statsmodels.graphics.tsaplots import plot_acf , plot_pacf

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

from skforecast.ForecasterAutoreg import ForecasterAutoreg
from skforecast.ForecasterAutoregCustom import ForecasterAutoregCustom
```

```
from skforecast.ForecasterAutoregMultiOutput import ForecasterAutoregMultiOutput
from skforecast.model_selection import grid_search_forecaster
from skforecast.model_selection import backtesting_forecaster

from joblib import dump, load

#      OTHERS
# ============================== #
from sklearn.inspection import permutation_importance
import multiprocessing

from python.funcionesPropias import *

#      WARNINGS
# ============================== #
import warnings
warnings.filterwarnings("ignore")
import dill
```

[ ]:
```
lineasMetro = pd.read_csv('../metroAnalisis/data/lineasMetro.csv' , encoding=␣
↪'latin-1')
```

[ ]:
```
lineasMetro['Línea'] = lineasMetro.index
lineasMetro.fillna('-', inplace= True )
lineasMetro['Terminales'] = lineasMetro['Terminales'].apply(lambda x : '-'.
↪join(x.split('\x96'))  if len(x.split('\x96')) >1 else x )
lineasMetro.drop(['Gálibo','Andén'], axis = 1, inplace= True)
```

2 Motivación

El Metro de Madrid fué inagurado en 1919 por el rey Alfonso XIII, aquella primera "red" de Metro constaba únicamente con ocho paradas, desde la Puerta del Sol hasta Cuatro Caminos. Tuvo tal éxito el nuevo medio de transporte en la ciudad que fué usado por más de 14 millones de usuarios.

Actualmente el Metro de Madrid, es la segunda red de metro mas extensa de la Unión Europea y la cuerta del mundo, consta de 13 líneas con 278 paradas distribuidas por toda la ciudad, creando una gran red de transporte de casi 290km, estableciendo la red de transporte más eficiente de la capital.

[ ]:
```
display(lineasMetro.set_index('Línea'))
print('Tabla extraída de wikipedia')
```

Línea	Terminales	Longitud	Estaciones
0	Pinar de Chamartín - Valdecarros	20,8 km	31
1	Las Rosas - Cuatro Caminos	14 km	20
2	Villaverde Alto - Moncloa	16,4 km	18
3	Argüelles - Pinar de Chamartín	16 km	23

4	Alameda de Osuna - Casa de Campo	23,2 km	32
5	Circular	23,5 km	28
6	Hospital de Henares - Pitis	31,2 km	29
7	Nuevos Ministerios - Aeropuerto T4	16,5 km	8
8	Paco de Lucía - Arganda del Rey	38,0 km	26
9	Hospital Infanta Sofía - Puerta del Sur	39,9 km	31
10	Plaza Elíptica - La Fortuna	5,3 km	7
11	MetroSur (Circular)	40,7 km	28
12	Ópera - Príncipe Pío	1,1 km	2
13	-	285,1 km	278

Tabla extraída de wikipedia

En la Comunidad de Madrid (CAM) el transporte público preferido por los Madrileños es el Metro y en los tiempos que nos encontramos (de pandemia), resultaría interesante estimar el volumen de pasajeros que recibirá el Metro en distintos instantes del tiempo.

Apoyados en la temperatura, viento, presión atmosférica y cantidad de rayoUV, vamos a intentar predecir la cantidad de viajeros en el metro de manera mensual (debido a que no he encontrado datos diarios, o incluso por horas).

[ ]:
```
df = pd.read_csv('./data/dfFinal.csv')
print(f"Tenemos {df.shape[1]} columnas y {df.shape[0]} filas")
df.drop(['mes_x', 'ano_x','mes_y', 'ano_y','festividad'], axis=1, inplace=True)
```

```
Tenemos 25 columnas y 396 filas
```

[ ]:
```
df.columns = ['fecha','volumenMetro', 'tmed', 'prec', 'tmin', 'tmax', 'dir',
       'velmedia', 'racha', 'presMax', 'presMin', 'tmedStd', 'precStd',
       'tminStd', 'tmaxStd', 'dirStd', 'velmediaStd', 'rachaStd', 'presMaxStd',
       'presMinStd']

df['fecha'] = pd.to_datetime(df['fecha'], format='%d/%m/%Y')
df = df.set_index('fecha')
df = df.asfreq('MS')
df = df.sort_index()
df = df.query("'01/01/2000' < fecha < '01-01-2020'")
display(df.head()[['volumenMetro','tmed','prec','racha']])
```

fecha	volumenMetro	tmed	prec	racha
2000-02-01	92.705	10.748276	5.448276	951.093103
2000-03-01	102.479	12.080645	6.777419	944.209677
2000-04-01	83.902	10.683333	6.273333	935.190000
2000-05-01	94.966	17.993750	12.743750	939.990323
2000-06-01	93.300	24.047917	16.800000	943.486667

Hemos recogido los datos desde Enero de 2000 hasta Diciembre de 2019 de volumen de pasajeros, datos climatológicos como temperatura, viento y presión medias y sus desviaciones típicas menusuales. Los datos han sido extraidos del banco de datos del ayuntamiento de Madrid y de la AEMET.

Hemos tenido que realizar una imputación de algunos de los datos, pues había datos nulos de presión atmosférica. El método utilizado, al tratarse de una serie temporal, ha sido la interpolación que ofrece la librería pandas.

[ ]:
```
df = df.interpolate(method='time')
```

3 EDA

Lo primero que debemos hacer es un pequño análisis exploratorio de las variables input y de la variable objetivo. Decimos que una variable objetivo, cuando es la elegida para estimar y las variables input, son las que se basará nuestro modelo para realizar las predicciones. En nuestro conjunto de datos, tenemos diferenciadas tres tipos de variables.

Variable de tiempo(fecha), hasta el año 2020 ya que debido a la pandemia, los datos se han visto alterados de manera muy fuerte y tener un frecuencia mensual, no tendríamos datos suficientes para hacer una buena estimación, es por esto, que se ha optado recurir a datos hasta el 2020, también a tener en cuenta esta frecuencia mensual, esto se debe a que los datos que disponemos sobre el volumen de pasajeros en el metro de Madrid es mensual. En una puesta en producción los datos podrían ser en streaming, con una actualización de minutos, probablemente gestionado en un entorno spark o con un procesado en bach. Hemos propuesto una pequeña base de datos SQL con tres tablas, que contienen el maestro de fechas, donde determina si es laborable, festivo, fin de semana… Otra tabla con los datos meteorológicos, agrupados por la variable tiempo, aunque en la ciudad de Madrid, por ejemplo existen 3 estaciones disponibles de donde sacar datos, estos datos los facilita la AEMET con su API y contiene un apartado para desarrolo de aplicaciones en streaming. Y por su puesto el volumen de pasajeros, que mediante los tornos ya instalados en las paradas de Metro se podría llevar el conteo casi en tiempo real. Por lo que, aunque la aplicación que propongo está gestionada en mensual, se podría extrapolar a un entorno en streaming o incluso para análisis en bach.

El modelo propuesto es un modelo ARIMA, ya que se ha comprobado en diversos estudios que para variables como la que vamos a estudiar tiene un gran rendimiento, además de su fácil interpretación. Además, al encontrarnos ante una serie temporal con carácter estacionario va el método clásico ARIMA tendrá un gran rendimiento.

3.1 Volumen de pasajeros

Estos datos han sido extraidos de Banco de datos del ayuntamiento de Madrid el dato viene informado miles de viajeros que están registrados en la agencia de viajeros de la CAM.

[ ]:
```
display(df.volumenMetro.describe().to_frame().T)
```

```
              count        mean        std     min      25%     50%      75%  \
volumenMetro  239.0   88.589019  12.173598  48.479  84.0535  91.535  96.3655

                  max
volumenMetro  109.412
```

En el gráfico vemos como nos encontramos ante una serie casi en su totalidad estacionaria, hemos tenido que lidiar con una mala extracción de los datos, pues en el año 2010 aparecía una gran caida

de pasajeros. La decisión ha sido utilizar la media del año anterior y posterior, es decir, la media entre los datos de 2019 y 2021.

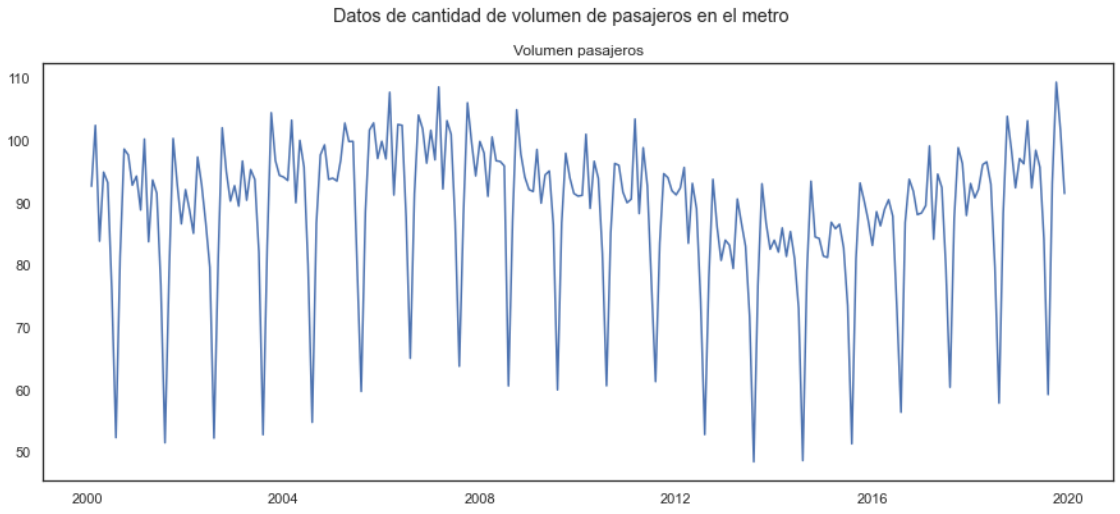
`Definición: Decimos que un proceso estocástico es estarionario en el sentido estricto cuando la`

Para nuestro estudio, nos vasta que el proceso sea estacionario en sentído débil, es decir

$$\begin{cases} \mu_t = \mu \quad \forall t \\ \sigma_t^2 = \sigma^2 \quad \forall t \\ Cov(X_t, X_{t+k}) = E[(x_t - \mu)(x_{t-k} - \mu)] = \gamma_k \quad \forall k \end{cases}$$

Esto quiere decir, que tanto media como varianza permanecen constantes con el tiempo y la covariancia, entre dos variables de la serie depende sólo de su separación en el tiempo.

[ ]:
```
fig, (ax1) = plt.subplots(1,1,figsize=(15,6))
fig.suptitle("Datos de cantidad de volumen de pasajeros en el metro")
ax1.plot(df.volumenMetro)
ax1.set_title('Volumen pasajeros')
fig.show()
```



Vemos como la serie presenta una cierta periodicidad de los datos, pues la forma que toma la serie es similar en todo el tiempo. Presentando cierta tendencia sinusoidal, si queremos aplicar el modelo ARIMA, debemos "eliminar" esta componente estacional, mediante por ejemplo normalizando podríamos obtener un ruido blanco.

Decimos que un proceso estocástico es un ruido blanco si

$$E[X(t)] = 0 \qquad V[X(t)] = \sigma^2 \qquad \gamma_k = 0$$

El objetivo es ajustar el modelo ARIMA es que el error producido sea un ruido blanco, esto significará que nuestro modelo está bien ajustado.

Actualmente, vemos con un gráfico de cajas y bigotes y un histográma, que la distribución de nuestra serie no es simétrica y esto se debe a que aunque tenemos una periodicidad de los datos muy clara, existe tendencia, provocando que la distribución de la serie no sea simétrica, con una media distante a la mediana y una desviación tipìca elevada. Por lo que nuestra serie no es estacionaria.

[ ]:
```
fig, (ax1,ax2) = plt.subplots(1,2, figsize=(12,4))
ax1.boxplot(df['volumenMetro'])
ax1.text(1,35, "Fig 1.: Gráfico boxplot e histograma del volumen de Pasajeros␣
 ↪del metro")
ax1.set_xticklabels('')
ax2.hist(df['volumenMetro'])
fig.show()
```

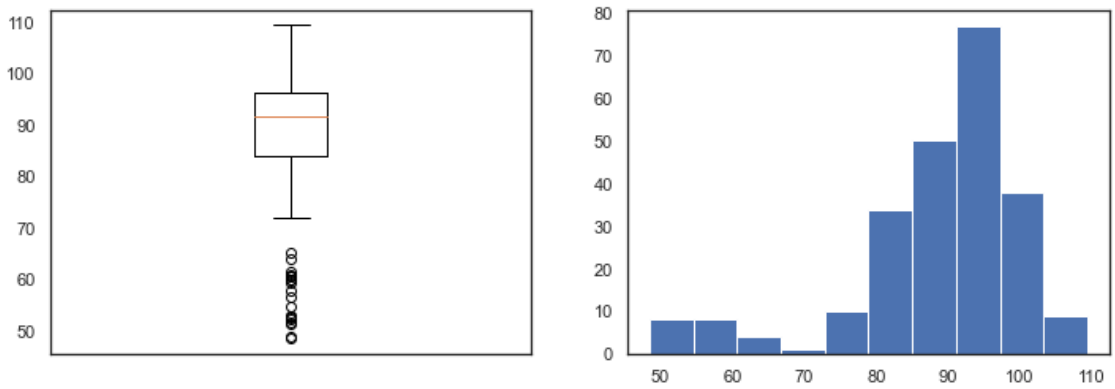
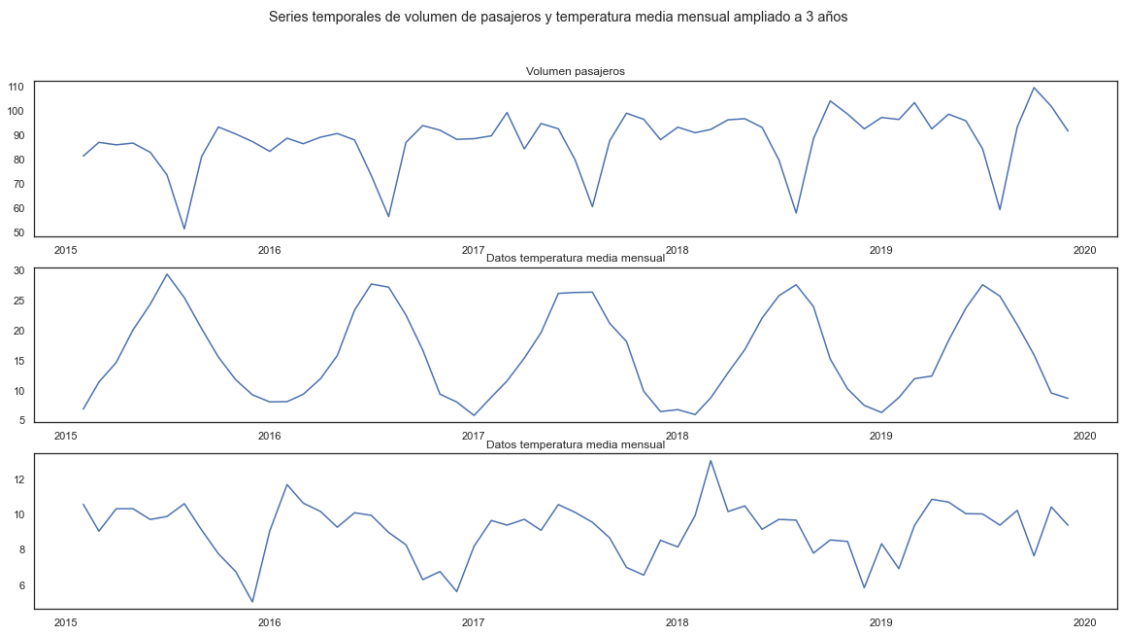


Fig 1.: Gráfico boxplot e histograma del volumen de Pasajeros del metro

3.2 Variables climatológicas

Vamos a exponer la comparación de un par de series de la variable objetivo con alguna exogena de apoyo, aunque el estudio se ha realizado en todas las variables. Vamos a aplicar un filtro de fecha de 5 años para poder comprar mejor.

[ ]:
```
fig, (ax1,ax2,ax3) = plt.subplots(3,1,figsize=(20,10))
fig.suptitle("Series temporales de volumen de pasajeros y temperatura media␣
 ↪mensual ampliado a 3 años")
ax1.plot(pd.Series(df.query("fecha > '01-01-2015'").volumenMetro))
ax1.set_title('Volumen pasajeros')
ax2.plot(pd.Series(df.query("fecha > '01-01-2015'")['tmed']))
ax2.set_title("Datos temperatura media mensual")
ax3.plot(pd.Series(df.query("fecha > '01-01-2015'")['velmedia']))
ax3.set_title("Datos temperatura media mensual")
fig.show()
```



Vemos como claramente en las épocas de mas calor, hay menos usuarios de Metro que en los meses más fríos. Parece que la temperatura media puede ayudarnos a estimar. Si nos fijamos en el viento sin embargo, cuando se producen disminuciones en la valocidad del viento, vemos como en general, se producen los aumentos de volumen de pasajeros. Esto es debido a que parece que va con un ligero retraso en su periodicidad, ya que en julio/agosto se produce un aumento de la temperatura, con una disminución del volumen de pasajeros, pero el viento se mantiene constante para estos meses. Es a partir de septiembre cuando se produce esta disminución de la velocidad media del viento y una disminución de la temperatura y el aumento del volumen de pasajeros.

4 Modelo ARIMA

Destacamos los modelos clásicos: - Decimos que un modelo es AR(p) (Autoregresivo de orden p), cuando las autocorrelaciones simples decrecen de manera exponencial y existen p autocorrelaciones distintas de 0. - Decimos que un modelo es MA(q) (Medias móviles), cuando las autocorrelaciones simples decaen y se cortan de forma rápida, sin embargo las autocorrelaciones parciales decrecen exponencialmente. - Decimos que un modelo es ARMA(p,q), cuando comparten las características de ambos modelos.

Definimos modelo ARIMA (Autoregresivo integrado de medias móviles), como el modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para una predicción hacia el futuro. Como ya hemos explicado, las series estacionarias son las que tienen media 0, por lo tanto, un proceso no estacionario lo llamaremos proceso integrado si al hacer una diferenciación se obtienen procesos estacionarios.

Decimos que aplicamos una diferenciación de orden k a una serie, cuando teniendo X_t le restamos la observación de $k $ instantes anteriores, es decir, $ X_{t-k} $.

Vamos a hacer una diferenciación de la serie para ver si conseguimos eliminar la media igual a cero

y así poder aplicar un modelo $ARIMA(p,d,q)(P,D,Q)_s$.

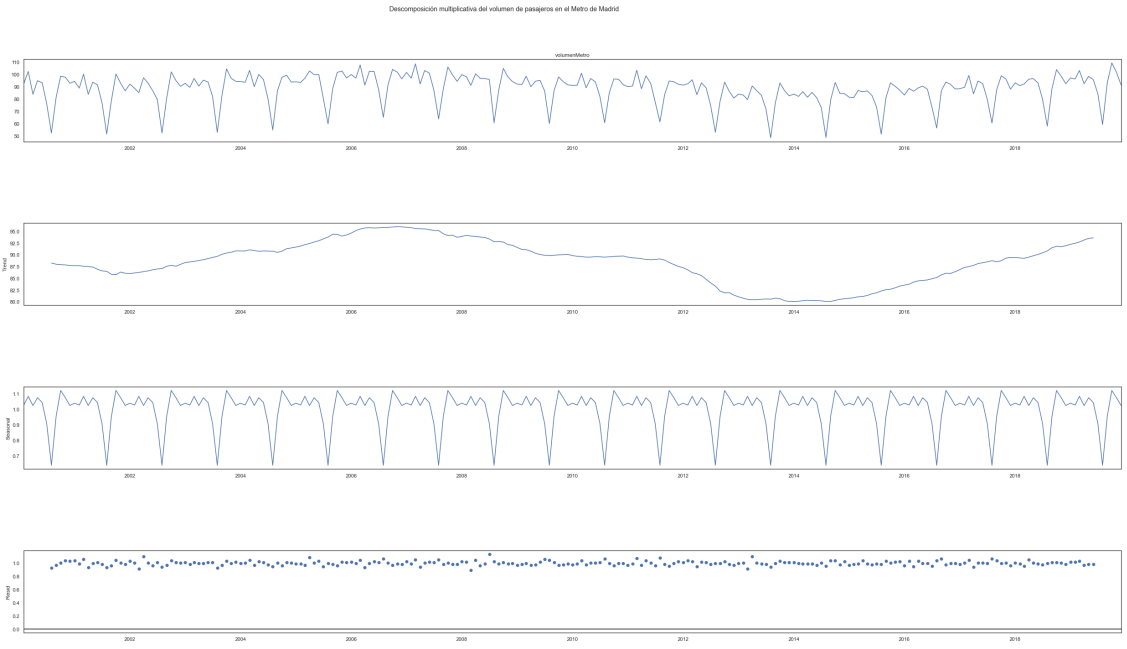
Vamos a utilizar la metodología Box-Jenkis para ajustar el modelo lo más fielmente posible, para realizar unas buenas predicciones.

[ ]:
```
print('Esquema de la metodología Box-Jenkins')
Image(url= "https://www.researchgate.net/profile/Alexander-Gorina-Sanchez/
 ↪publication/318445593/figure/fig1/AS:631670810099722@1527613465199/
 ↪Figura-1-Metodologia-de-Box-Jenkins-para-el-analisis-de-series-de-tiempo.png"
, width=500, height=600)
```

```
Esquema de la metodología Box-Jenkins
```

[ ]:
```
<IPython.core.display.Image object>
```

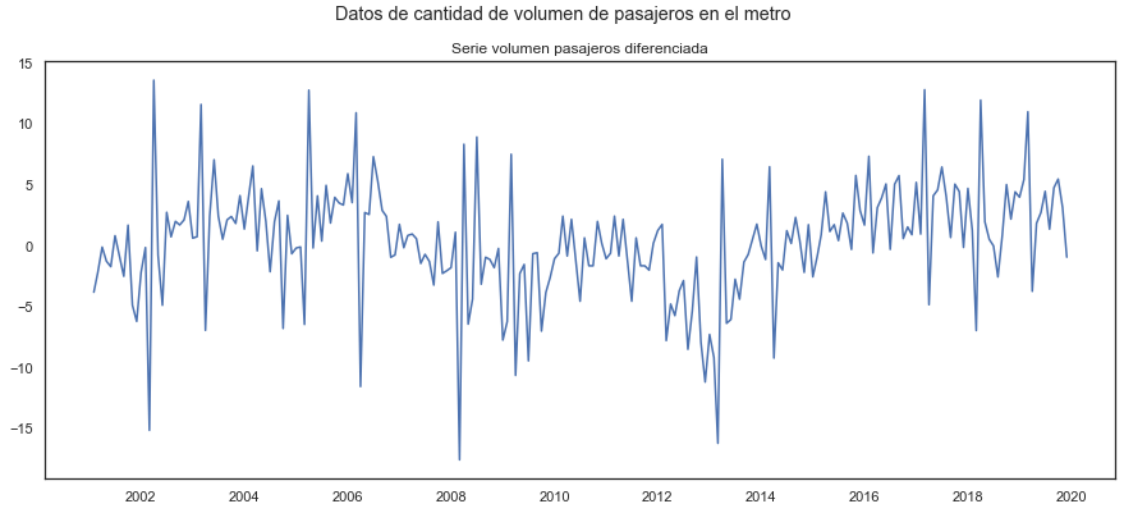
[ ]:
```
resultV=seasonal_decompose(df['volumenMetro'], model='multiplicable')
fig = resultV.plot()
fig.suptitle('Descomposición multiplicativa del volumen de pasajeros en el␣
 ↪Metro de Madrid')
fig.set_size_inches(40, 22)
fig.show()
```



[ ]:
```
volDiff = df.volumenMetro.diff(12)
```

[ ]:
```
fig, (ax1) = plt.subplots(1,1,figsize=(15,6))
fig.suptitle("Datos de cantidad de volumen de pasajeros en el metro")
ax1.plot(volDiff)
```

```
ax1.set_title('Serie volumen pasajeros diferenciada')
fig.show()
```



5 Bibliografía

https://www.ucm.es/data/cont/docs/518-2013-11-11-JAM-IAST-Libro.pdf