



Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Práctica 1

Ing. Magdalena Bouza, Ing. Carlos German Carreño Romano

Encuesta

- Poll: https://docs.google.com/forms/d/e/1FAIpQLSfxOrBbJpef7KZmoOdHE3CabA5j2843OMP1Ft-k3glaERB-Rg/viewform?usp=pp_url

Agenda

1. Variables aleatorias
2. Gráficos
3. Datos de procesos estocásticos
4. Series de tiempo
5. Repo

Análisis de Series de Tiempo

Box and Jenkins plantean tres etapas de **modelado**:

1. Especificaciones

- a. Gráficos
- b. Estadísticas
- c. Contexto

2. Ajuste

- a. Parámetros
- b. Valores

3. Diagnóstico

- a. Testing

“everything should be made as simple as possible but no simpler”

$$X \sim \mathcal{N}(\mu, \sigma) \rightarrow f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

$$f(t) = \frac{1}{\sqrt{\nu\pi} \text{B}(\frac{1}{2}, \frac{\nu}{2})} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Variables aleatorias



```
import numpy as np

# random values data series
X = np.random.normal(mu, sigma, N)
Y = np.random.standard_t(degFreedom, N)
```

```
>>> X
array([ 0.04308072, -0.08878357,  0.07584408, ...,  0.01327434,
        0.05514431,  0.0305423 ])
```

```
>>> Y
array([ 0.62984362,  0.24089912,  0.95562105, ..., -0.70564368,
        0.00498332, -1.23120948])
```

```
>>> 
```

```
import matplotlib.pyplot as plt
```

```
# Plots
```

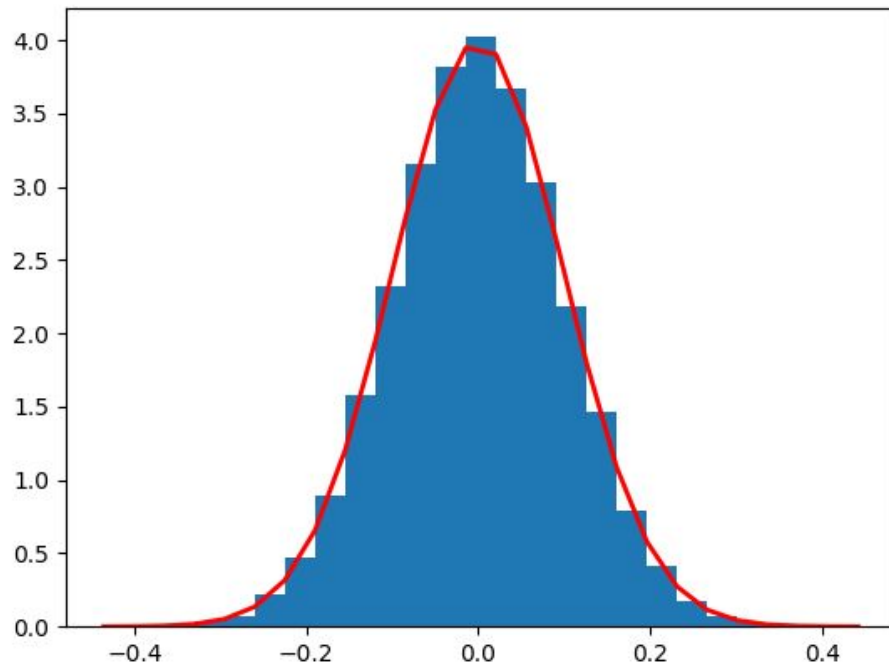
```
plt.plot(X)
```

```
plt.show()
```

```
count, bins, ignored = plt.hist(X, bins, density=True)
```

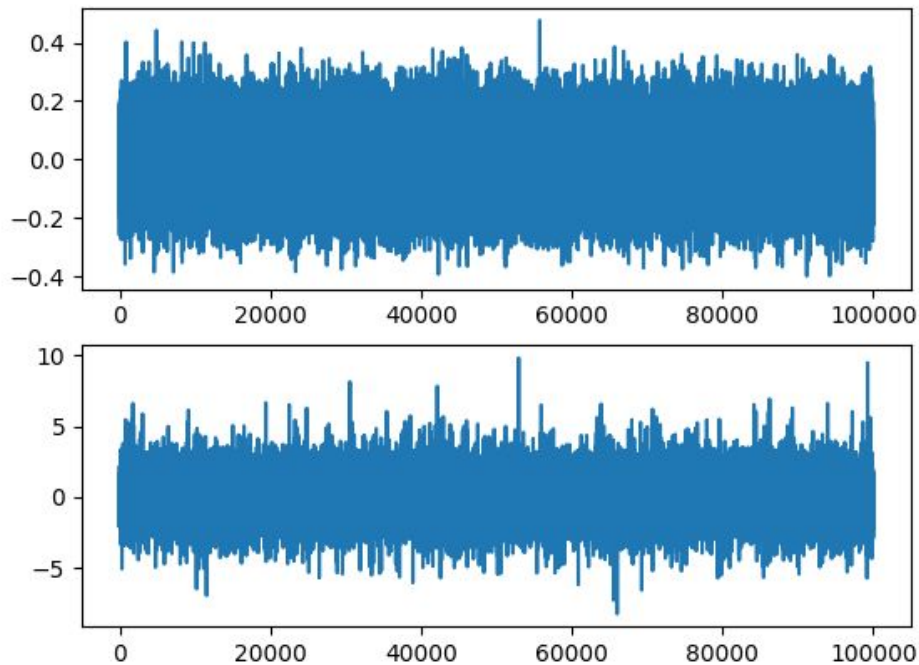
```
plt.plot(bins, 1/(sigma*np.sqrt(2*np.pi))*np.exp(-(bins-mu)**2/(2*sigma**2)),  
linewidth=2, color='r')
```

```
plt.show()
```




```
## data series
```

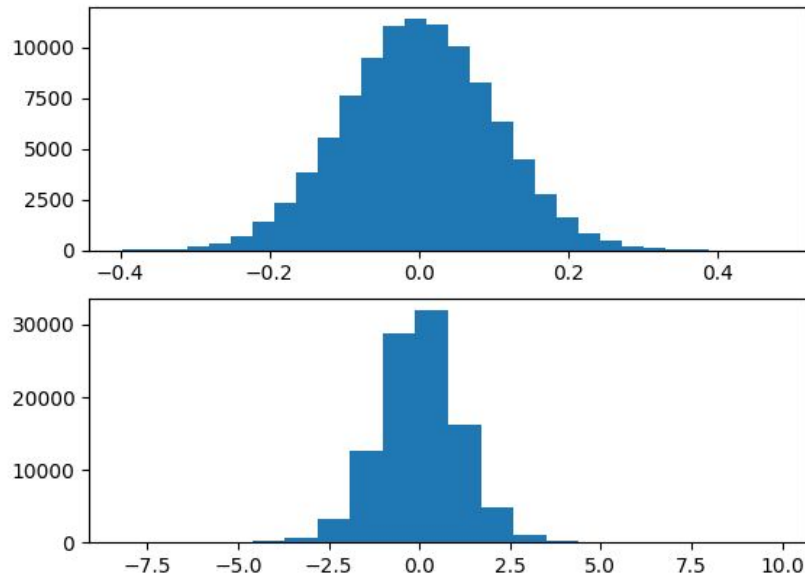
```
fig, (ax1, ax2) = plt.subplots(2)  
fig.subtitle('X vs. Y timeseries')  
ax1.plot(X)  
ax2.plot(Y)  
plt.show()
```



Gráficos



```
## histograms
fig, (ax1, ax2) = plt.subplots(2)
fig.subtitle('X vs. Y histograms')
ax1.hist(X, bins=30)
ax2.hist(Y, bins=20)
plt.show()
```



Tablas descriptivas



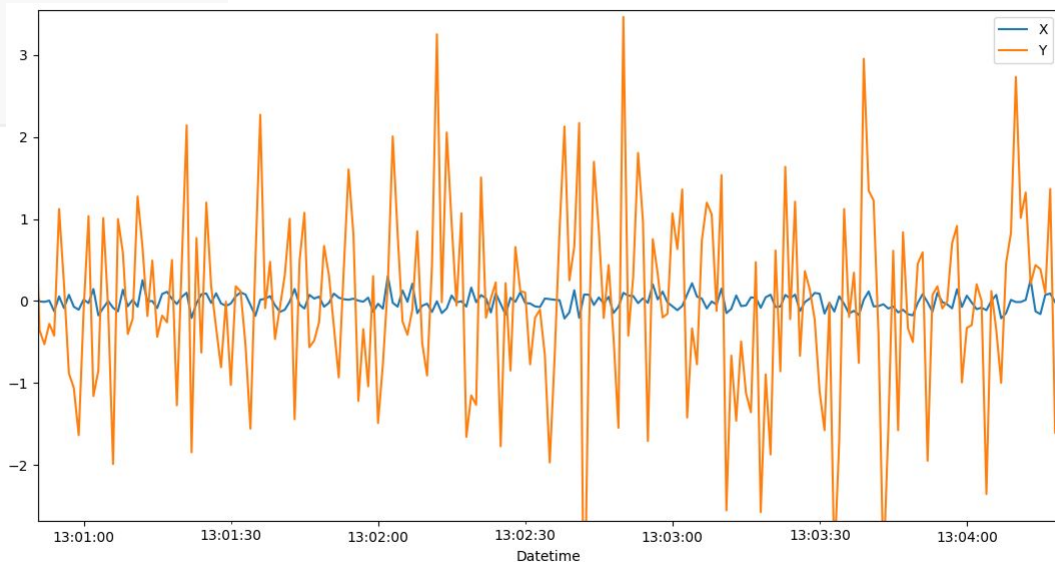
```
## stats
df.X.describe
df.Y.describe()
dt = pd.DataFrame(df.X.describe())
dt=pd.concat([df.X.describe(), df.Y.describe()],axis=1)
```

Stat	X	Y
count	100000	100000
mean	6.733E-05	6.608E-03
std	1.001E-01	1.129E+00
min	-3.980E-01	-8.216E+00
25%	-6.742E-02	-6.949E-01
50%	-3.210E-05	4.958E-03
75%	6.800E-02	7.056E-01
max	4.761E-01	9.797E+00

Series de tiempo



```
# adding datetime index
time = pd.date_range('2021-10-08', periods=N, freq='s')
df.insert(col, "Datetime", time, True)
df.set_index('Datetime')
df.set_index('Datetime').plot()
plt.show()
```



Repo

<https://github.com/charlieromano/TimeSeries>



- Datasets
- Docs
- Pics
- Scripts
- README

charlieromano / TimeSeries Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Code

About
Este es el repo de la materia de Análisis de Series de Tiempo
Readme
BSD-2-Clause License

Releases
No releases published

Packages
No packages published


Languages
Python 100.0%

File	Commit	Time
Datasets	practica 01	39 seconds ago
Docs	update README	3 months ago
Pics	practica 01	39 seconds ago
Scripts	practica 01	39 seconds ago
LICENSE	Initial commit	4 months ago
README.md	update README	3 months ago
STEM-RNN.Rmd	agrego datasets para actividad 1	2 hours ago

README.md

TimeSeries

Este es el repo de la materia de Análisis de Series de Tiempo



Hands-on

Actividad

1. Generar una serie de tiempo con valores aleatorios y graficarla.
2. Graficar una serie de tiempo de los ejemplos del repositorio: <https://github.com/charlieromano/TimeSeries>
3. Redactar un informe describiendo series de tiempo distintas (máx. 3 páginas)
4. Crear una base de datos SQL con las tres series elegidas.