الفصل الثالث

التشويش الأبيض و السير العشوائي

White Noise and Random Walk

محتويات الفصل الثالث

- مقدمة Introduction
- التشويش الأبيض White Noise
- رسم السلاسل الزمنية Plotting Time Series
- مقارنة المتوسط الحسابي والانحراف المعياري بمرور الوقت
- فحص مخططات الارتباط التلقائي Autocorrelation Plots
- السير العشوائي Random Walk
- ملخص Summary

لتحميل ملحقات هذا الفصل من الأكواد البرمجية والشروحات التوضيحية زيارة المستودع التالي على صفحتي في موقع GitHub :

- السلاسل الزمنية باللغة العربية

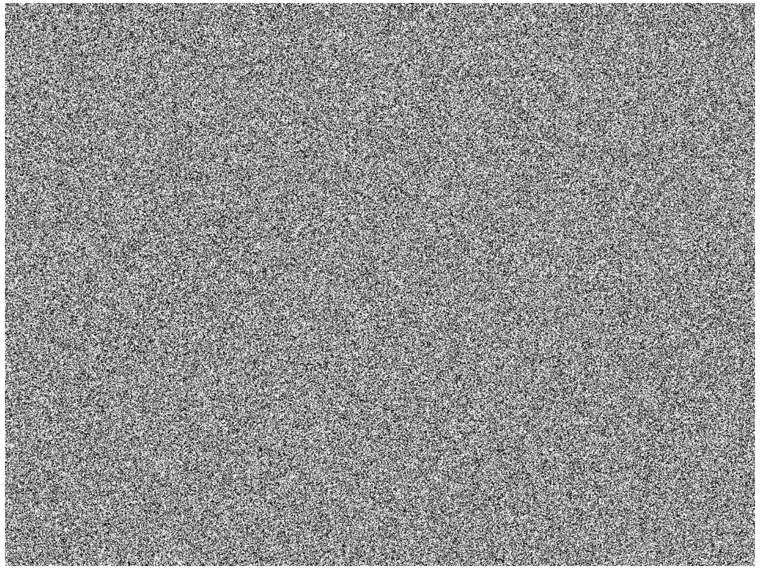
مقدمة Introduction

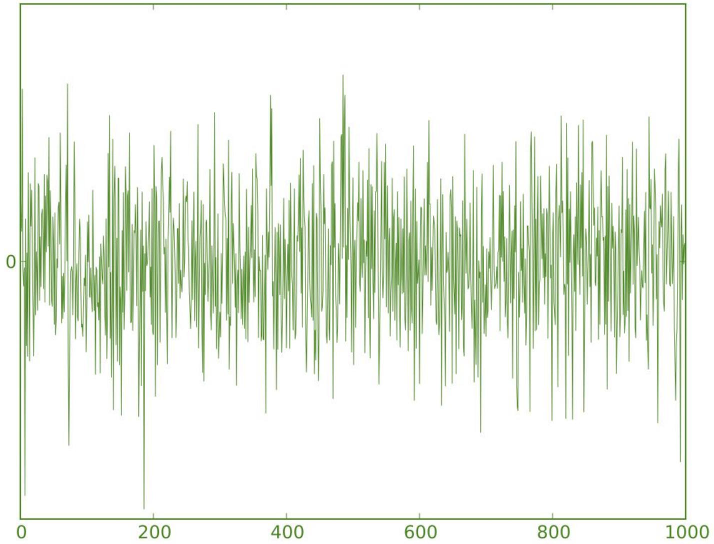
في بعض الأحيان يمكن أن يكون التنبؤ ببيانات السلاسل الزمنية Time Series Forecasting مهمة صعبة. قبل أن تبدأ حتى في عملية التنبؤ يجب أن تسأل نفسك السؤال التالي - هل يستحق الأمر إجراء التنبؤ؟ حسنًا ، إذا كانت بياناتك تشبه التشويش الأبيض White Noise أو السير العشوائي Random Walk ، فإن الإجابة ببساطة هي لا !!! ستتعلم اليوم الأفكار الكامنة وراء هذين الموضوعين الأساسيين في تحليل السلاسل الزمنية و سنراجع بعض النظريات ونلقي نظرة عملية من خلال بايثون.

التشويش الأبيض White Noise

يعد مفهوم التشويش الأبيض White Noise ضروريًا لفهم تحليل السلاسل الزمنية والتنبؤ بسلوكها. في أبسط الكلمات ، يخبرك التشويش الأبيض ما إذا كان يجب عليك نمذجة نموذج للتنبؤ أو تحسينه أم لا !

دعني أوضح ذلك . السلسلة الزمنية التي تصنف على أنها تشويش أبيض هي سلسلة غير قابلة للتنبؤ أو في أحسن الأحوال من الصعب التنبؤ بها بسبب أنها سلسلة من الأرقام العشوائية لا تحتوي على أنما Patterns وكما نعلم كلما زادت العشوائية (الفوضى) Entropy في النظام صعب التنبؤ بسلوكه.





التشويش الأبيض White Noise

يجب استيفاء الشروط التالية حتى يتم تصنيف السلسلة الزمنية على أنها تشويش أبيض :

- المتوسط الحسابي mean لقيم السلسلة هو صفر أو تقريبًا صفر
- الانحراف المعياري Standard Deviation ثابت - لا يتغير بمرور الوقت
- العلاقة بين السلسلة الزمنية ونسختها المتأخرة ليست ذات دلالة

قد يصعب فهم هذه النقاط وخصوصًا النقطة الأخيرة نظرًا لأننا لم نستكشف بعد مفهوم الارتباط التلقائي Autocorrelation وهو تحديد ما إذا كان هناك ارتباط كبير موجود بين السلسلة الزمنية الحالية ونفس السلسلة الزمنية التي تم إزاحتها بفترات N سنشرح هذه المفاهيم بشكل أوسع بعد قليل !

توجد ثلاث طرق (سهلة) لاختبار ما إذا كانت السلاسل الزمنية تشبه التشويش الأبيض :

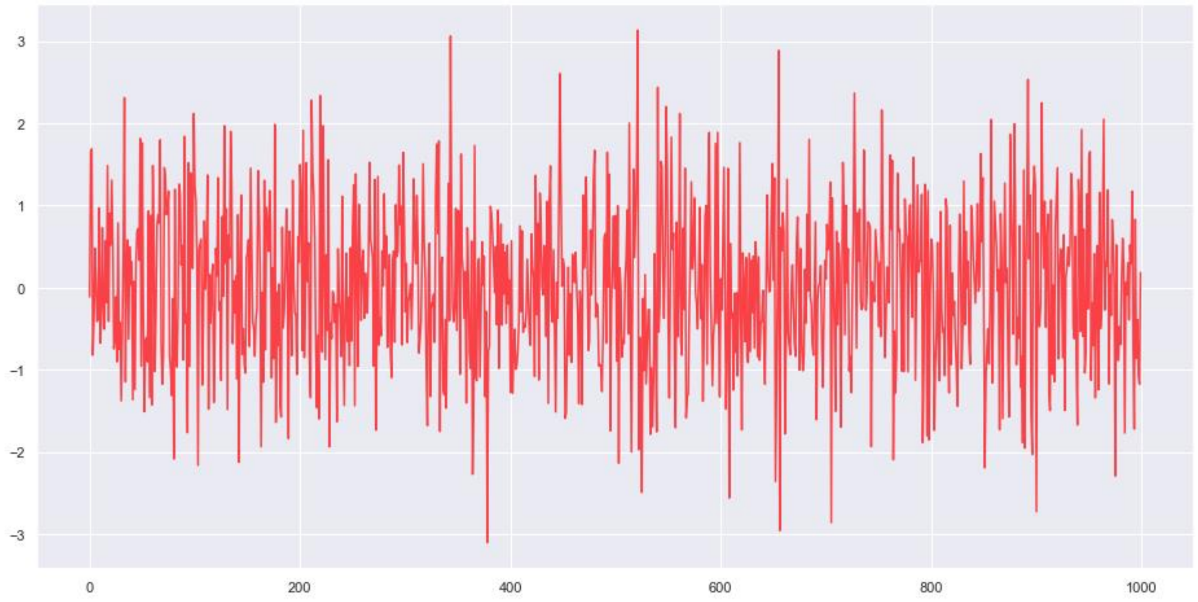
1. من خلال رسم السلسلة الزمنية بيانيًا
2. من خلال مقارنة المتوسط والانحراف المعياري بمرور الوقت
3. من خلال فحص الرسوم البيانية للارتباط التلقائي Autocorrelation plots

١- رسم السلاسل الزمنية Plotting Time Series

هذه هي الطريقة الأسهل. الهدف هو رسم السلسلة بأكملها والتأكد بالملاحظة البصرية على أن متوسط القيمة هو صفر ، وأن الانحراف المعياري ثابت بمرور الوقت ، وأنه لا توجد أنماط مميزة مرئية.

سنحاكي سلسلة زمنية تحتوي على بيانات عشوائية (تشويش أبيض) عن طريق مكتبة NumPy وتصويرها بيانيًا !

```
# استيراد المكتبات
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# بيانات عشوائية لسلسلة زمنية عبارة عن تشويش أبيض
white_noise = np.random.randn(1000)
time = np.arange(len(white_noise))

# تفضيلات نمط التصوير البياني
sns.set_style("dark")
sns.set(color_codes=True)

# تهيئة التصوير البياني و أبعاده
ax, fig = plt.subplots(figsize=(16,8))

# تصوير السلسلة الزمنية
sns.lineplot(x=time, y=white_noise,
             color='#FF4343')
plt.show()
```



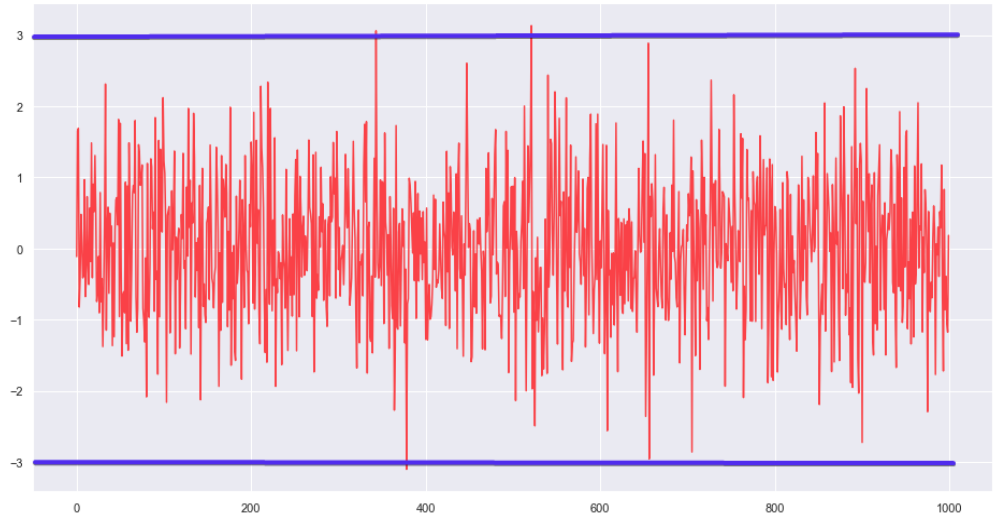
أول ما يتم ملاحظته هو عشوائية القيم أيضًا هناك بعض الارتفاعات والانخفاضات العرضية Spikes، لا توجد أنماط مرئية في السلسلة. ولكن ما يهم أكثر هو ملاحظة ما إذا كان المتوسط الحسابي مساويًا للصفر و الانحراف المعياري ثابتًا وهو ما سنرصده في الشريحة التالية .

١- رسم السلاسل الزمنية Plotting Time Series

نلاحظ أن التصوير البياني يحمل تماثلاً بالقيم على المحور y أي بشكل تقريبي كل قيمة موجبة يوجد ما يقابلها بالسالب وعند احتساب المتوسط الحسابي لهذه القيم فإن النتيجة ستكون صفراً أو مقاربة جدًا للصفر وهذا أول معيار للتشويش الأبيض White Noise

مثال عند إيجاد المتوسط الحسابي mean للقيم التالية [23, -23, 5,-5,7,-7]

Mean = [3, -3, 5,-5, 7,-7] /6 = 0



نلاحظ أن القيم في السلسلة ذات تباين ثابت constant variance أي أن القيم تتذبذب بين قيم محددة لا تخرج عن نطاقها ألا وهو -3 و 3 لا نرى رتمًا تصاعديًا أو تنازليًا للسلسلة مع مرور الزمن وهذا ما يدل على ثبات في الانحراف المعياري Standard deviation وبالتالي تحقق معيار آخر من معايير وجود التشويش الأبيض White Noise

2- مقارنة المتوسط الحسابي والانحراف المعياري بمرور الوقت

إذا افترضنا أن المتوسط الحسابي mean و الانحراف المعياري standard deviation لا يتغير بمرور الوقت لسلسلة زمنية ما ، فيجب أن تكون كلتا القيمتين متطابقتين تقريبًا لأي مجموعتين فرعيتين أي لو قمنا بتقطيع السلسلة الزمنية إلى أجزاء متساوية و قمنا بحساب المتوسط والانحراف لكل من هذه المجموعات فيجب أن تتساوى تقريباً قيم المتوسط والانحراف المعياري لجميع هذه المجموعات.

الهدف هنا هو تقسيم سلسلة التشويش الأبيض بشكل عشوائي إلى عدد من القطع (دعنا نقول 20 قطعة، لكل منها 50 عنصرًا) وحساب المتوسط والانحراف المعياري لكل منها. يمكنك تصور النتائج لتفسير أسهل.

```python
# تقسيم السلسلة إلى 20 جزء عشوائي بالتساوي زمنياً
white_noise_chunks =
np.split(white_noise, 20)


# حساب المتوسط الحسابي و الانحراف المعياري لكل جزء
means, stds = [], []

for chunk in white_noise_chunks:
    means.append(np.mean(chunk))
    stds.append(np.std(chunk))

# طباعة جميع المتوسطات والانحرافات المعيارية للمقارنة
print(f"means: {means}")
print("---" * 50)
print(f"standard deviations: {stds}")
```

```
means: [0.02836488045551929, -0.020656721224810815, -0.1471670410708862, 0.05314403110961476, -0.26114920190360463, 0.14793135600978524, -0.2738176078658923,
0.017401790479396075, 0.004424065874929159, 0.0832946050024743, -0.07240936945603504, -0.09043685338511968, 0.0247560880261877, -0.19292097570687375, -0.20961775499731566,
0.06823753946916859, -0.15172241847889695, -0.1328581389086227, 0.013532776440578487, 0.093260214222863 6]
------------------------------------------------------------------------------------------------------------------------------------------------------
standard deviations: [0.9489482628349547, 1.0783318098666825, 1.1427595468949245, 1.0942379259946031, 1.007697307762962, 0.7837824203501655, 0.9000208230634525,
1.0607780115777528, 1.009017682926548, 1.1332543820437475, 0.94955333718244, 0.869847749222071, 0.8743847985228957, 0.8310459073848016, 1.0036050369060536,
0.9974457362715776, 0.9046274156122556, 1.0585974791784851, 0.9580231912430538, 0.9790721429006451]
```

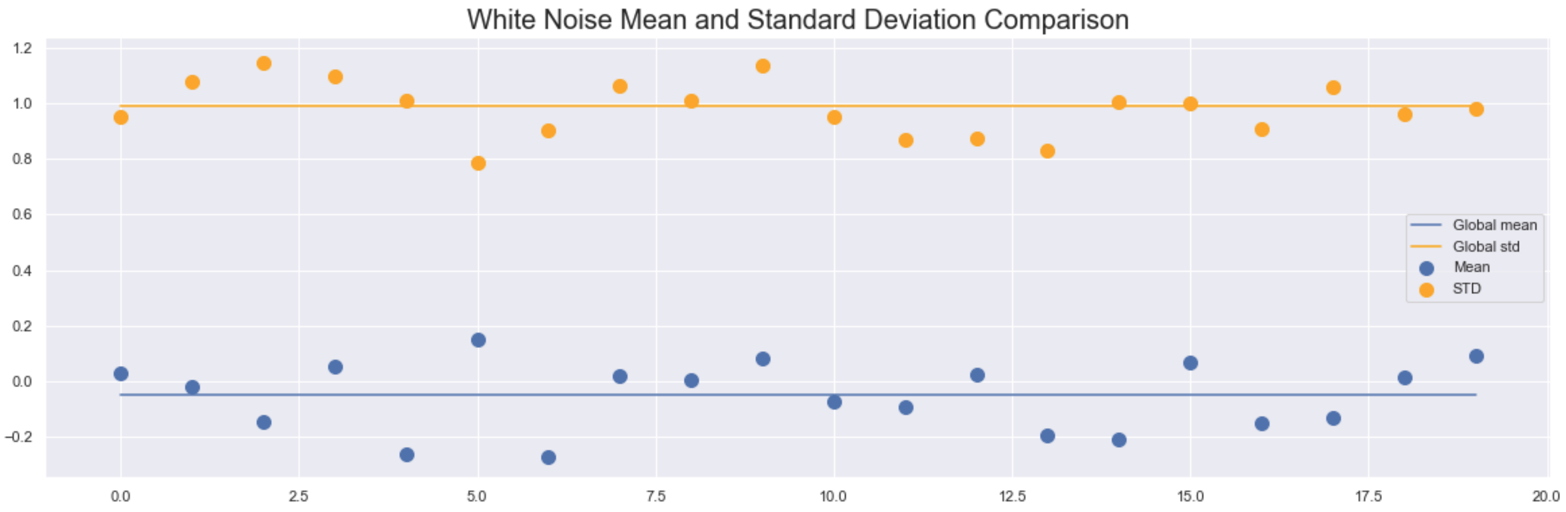
- نلاحظ أن قيم المتوسط الحسابي هي قريبة من الرقم 0
- نلاحظ أن قيم الانحراف المعياري قرية من الرقم 1

وهذا يدل على متوسط حسابي مقارب للصفر و انحراف معياري شبه ثابت على طول السلسلة الزمنية وهو من معايير التشويش الأبيض.

2- مقارنة المتوسط الحسابي والانحراف المعياري بمرور الوقت

فيما يلي تصوير بياني للمتوسط الحسابي العام والانحراف المعياري العام لكامل السلسلة الزمنية مقارنة بالمتوسطات الحسابية والانحرافات المعيارية للأجزاء العشرين على طول السلسلة الزمنية !

```python
# مقارنة المتوسطات الحسابية والانحرافات المعيارية للسلسلة الزمنية

mean_time_axis = np.arange(len(means))
std_time_axis = np.arange(len(stds))

# تفضيلات التصوير البياني
plt.rcParams["figure.figsize"] = (20,6)
plt.title('White Noise Mean and Standard
Deviation Comparison', size=20)

# الرسم البياني للمتوسط العام للسلسلة الزمنية
plt.plot(mean_time_axis,
[white_noise.mean()] * len(means),
         label='Global mean', lw=1.5)
# الرسم البياني للمتوسطات الحسابية للأجزاء المقطعة
plt.scatter(mean_time_axis, y=means,
label='Mean', s=100)

# الرسم البياني للانحراف المعياري العام للسلسلة الزمنية
plt.plot(std_time_axis,
[white_noise.std()] * len(stds),
         label='Global std', lw=1.5,
color='orange')
# الرسم البياني للانحراف المعياري للأجزاء المقطعة
plt.scatter(std_time_axis, y=stds,
label='STD', color='orange', s=100)

plt.legend();
```



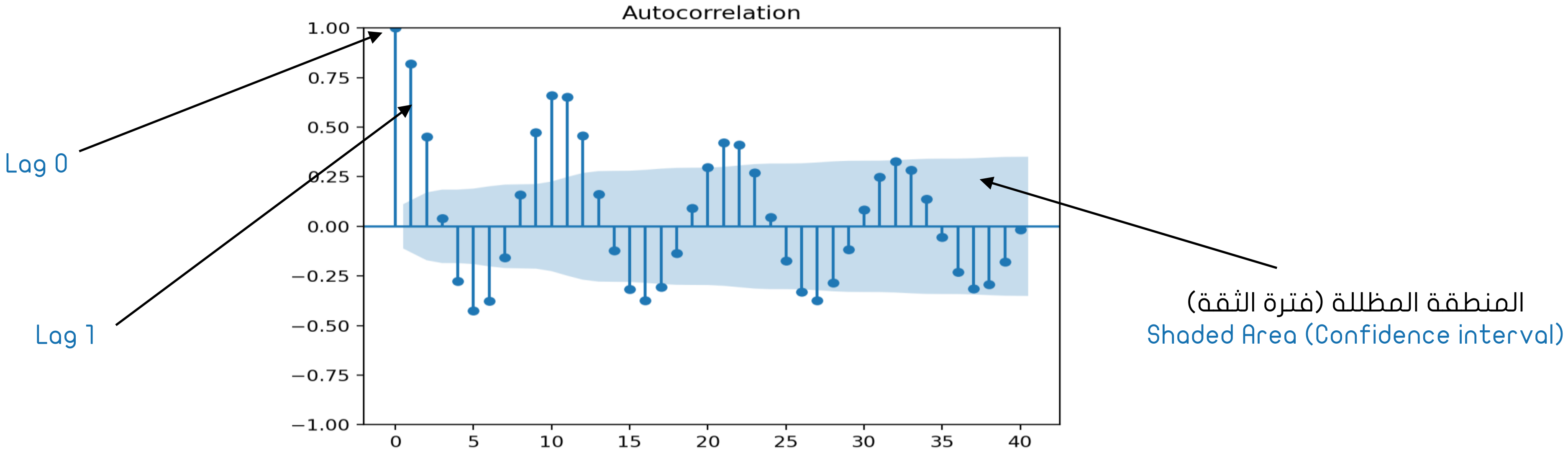
- نلاحظ أن قيم الانحراف المعياري للأجزاء المقطعة (النقاط الصفراء اللون) من السلسلة الزمنية مقاربة جدًا للانحراف المعياري (الكلي) للسلسلة الزمنية المتمثل بالخط الأصفر !

- الملاحظة ذاتها بالنسبة للمتوسط الحسابي للأجزاء المقطعة و المتوسط الحسابي الكلي حيث أنها قريبة للغاية من القيمة صفر !

3- فحص مخططات الارتباط التلقائي Examining Autocorrelation Plots

سنخصص فصلاً كاملاً عن كل ما يتعلق بمخططات الارتباط التلقائي Autocorrelation plots في هذه السلسلة ، لذلك لن نتعمق في الكثير من التفاصيل هنا. كل ما تحتاج إلى معرفته هو أن مخطط الارتباط التلقائي يُظهر الارتباط بين سلسلة زمنية مع نفسها ، متأخرةً بعدد محدد من الفترات lagged by a specific number of periods.

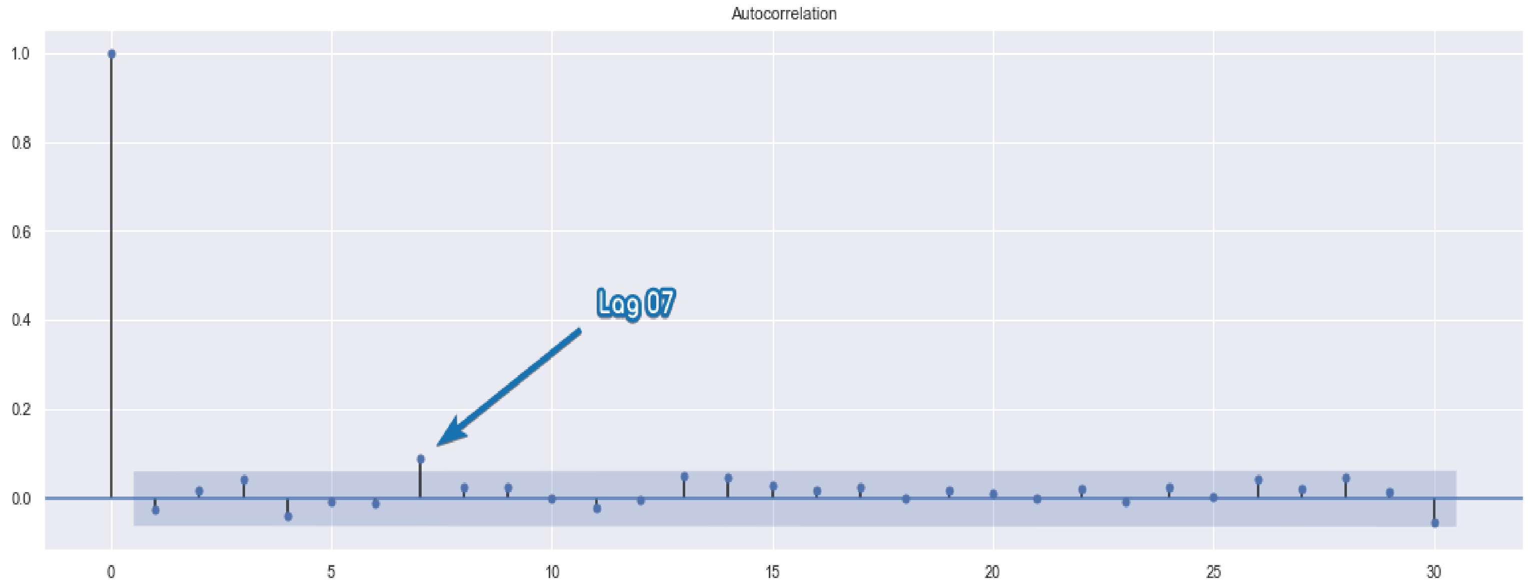
تتضمن مخططات الارتباط التلقائي أيضًا منطقة مظللة shaded area تمثل ما يسمى بفترة الثقة confidence interval. أي شيء داخل المنطقة المظللة يعني أن الارتباط في هذه النقطة ليس ذو دلالة إحصائية (ليس ذو أهمية كبيرة). سيكون للارتباط التلقائي في 0 واو أعلى ما يمكن أي بقيمة 1 ، بسبب أنه يتم حساب الارتباط بين سلسلتين زمنيتين متطابقتين (السلسلة الأصلية مكررة).



3- فحص مخططات الارتباط التلقائي Examining Autocorrelation Plots

يمكنك استخدام دالة plot_acf () من statsmodels للحصول على مخطط الارتباط التلقائي لسلسلة زمنية:

```
# استيراد المكتبات
import numpy as np
import statsmodels.api as sm

# هذه الدالة مسؤولة عن تثبيت توليد النتائج العشوائية
np.random.seed(0)

# سلسلة زمنية تمثل تشويش أبيض
white_noise = np.random.randn(1000)

# مخطط الارتباط التلقائي
sm.graphics.tsa.plot_acf(np.array(white_noise))
plt.show()
```



نلاحظ أن جميع الارتباطات التلقائية Autocorrelation الخاصة بال Lags تقع ضمن فترة الثقة (المنطقة المظللة) عدا Lag 07 والذي يعتبر بالكاد خارجًا منها كما هو موضح في المخطط البياني أعلاه ، أي أن جميع هذه الارتباطات ليست ذات أهمية إحصائية يعتمد عليها في التنبؤ وبالتالي ستعتبر هذه السلسلة تشويشًا أبيضًا White Noise.

السير العشوائي Random Walk

كما هو الحال مع السلسلة الزمنية ذات التشويش الأبيض White Walk لا يمكن التنبؤ بسلوك السلسلة التي تٌمثل مفهوم المشي العشوائي Random Walk ، و ما يجعلها مختلفة عن سلسلة التشويش الأبيض هي حقيقة أن القيم العشوائية ليست ناشئة بسبب الأرقام العشوائية إنما بسبب طريقة اعتماد القيمة الحالية على القيمة السابقة في السلسلة (سنسلط الضوء على ذلك بعد قليل).

لإنشاء سلسلة سير عشوائي وهمية :

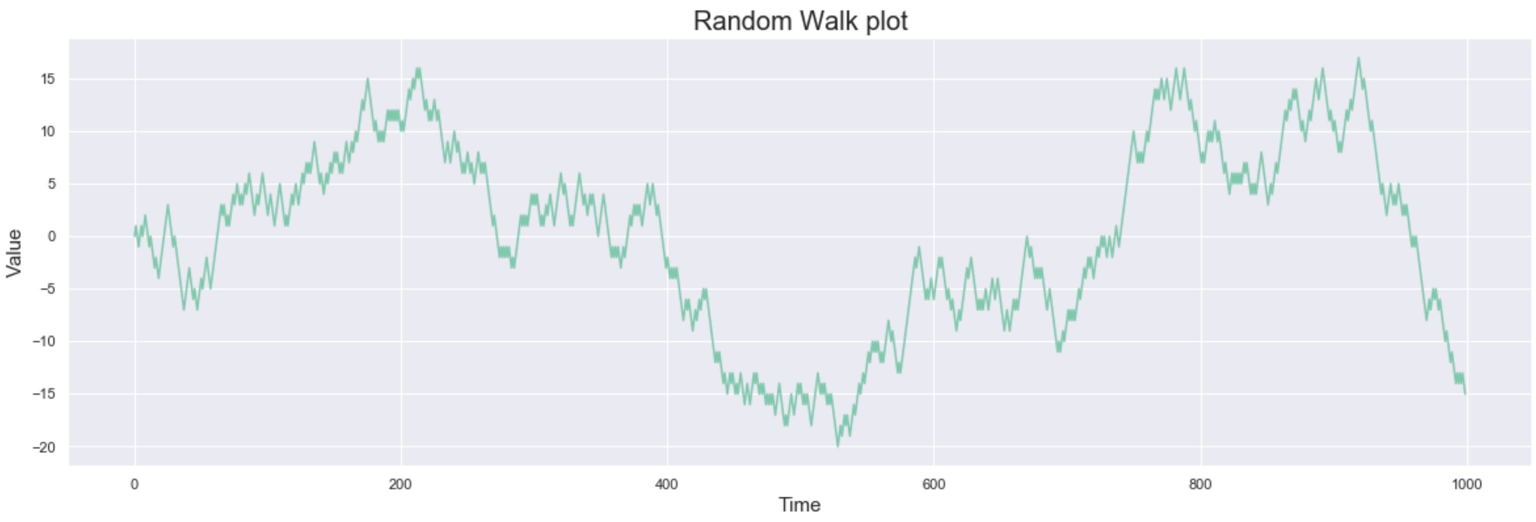
- ابدأ بقيمة عشوائية - دعنا نقول صفر.
- القيمة التالية هي القيمة السابقة بالإضافة إلى التباين العشوائي المضاف.

يمكنك تكرار عملية إضافة قيم إضافية عدة مرات كما تريد ستتعلم ذلك بشكل تطبيقي في الشريحة التالية!

السير العشوائي Random Walk

دعنا نطبق ما سبق عمليًا – سنبدأ بداية السلسلة بقيمة صفر ثم سنضيف 999 عنصرًا آخر كل عنصر سيعتمد على القيمة السابقة وسنضيف إليه أحد القيمتين ١ أو -١ بالاعتماد على الرقم العشوائي المتولد بواسطة خوارزمية تولد أرقاماً عشوائية فإذا كان الرقم أكبر من 0.5 سنضيف ١ و في حال كان أصغر من ذلك سنضيف -١ وبالتالي سيتم توليد سلسلة سير عشوائي غير قابلة للتنبؤ كما هو موضح في الأسطر البرمجية والصورة أدناه.

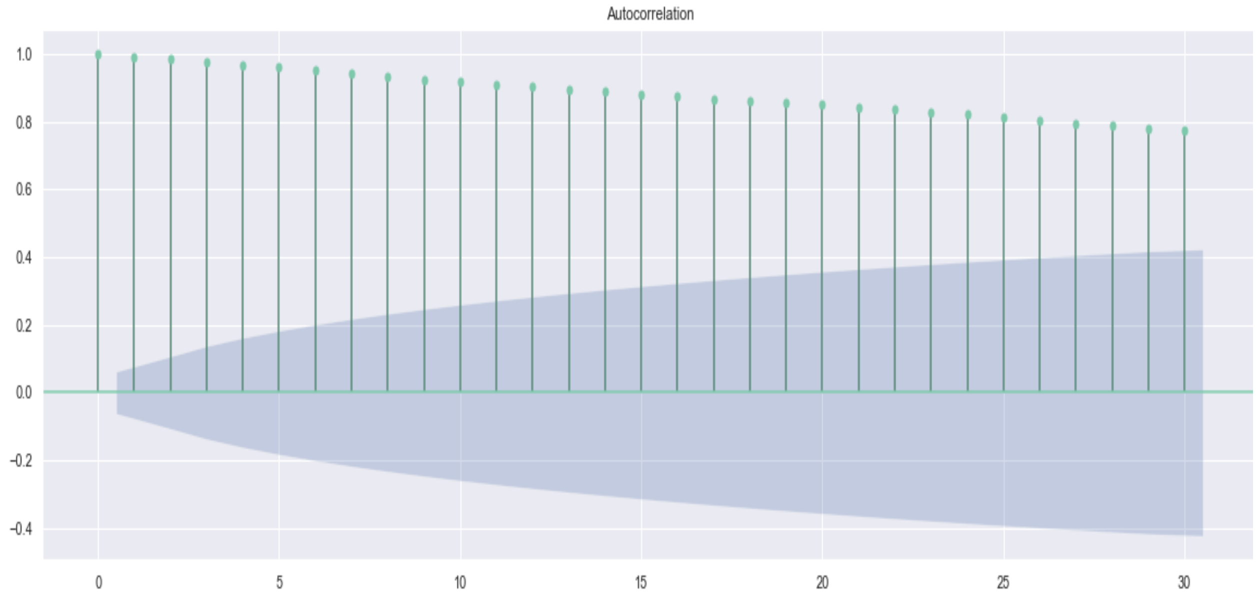
```python
# كما أشرنا سابقًا سنبدأ السلسلة بقيمة صفر
random_walk = [0]

np.random.seed(123)


for i in range(1, 1000):
    # اتجاه التحرك في السلسلة يعتمد على الرقم العشوائي المضاف
    if np.random.random() > 0.5:
        num = 1
    else:
        num = -1

    random_walk.append(random_walk[-1] + num)

# تصوير السلسلة الزمنية
g = sns.lineplot(x=np.arange(len(random_walk)),
                 y=random_walk, color='#79C9AB')

g.set_title("Random Walk plot", fontsize=20)
g.set_xlabel('Time', fontsize=15)
g.set_ylabel('Value', fontsize=15)
plt.show()
```



السير العشوائي Random Walk

الآن سنقوم باختبار الارتباط التلقائي على هذه السلسلة و نظرًا لوجود علاقة بين القيم الحالية والسابقة ، فإن مخطط الارتباط التلقائي سيبدو مختلفًا عن مخطط التشويش الأبيض White Noise !

```
sm.graphics.tsa.plot_acf(np.array(random_walk),
                         color='#79C9AB',
                         vlines_kwargs={"colors": '#508672'})
plt.show()
```

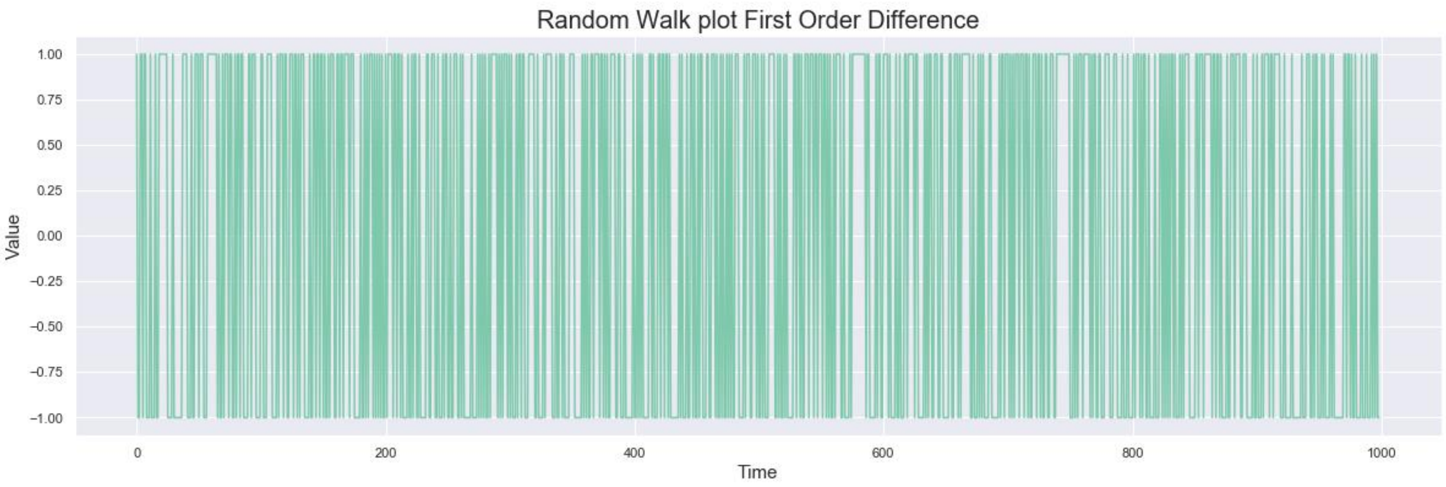


نلاحظ أن جميع الارتباطات التلقائية Autocorrelation الخاصة بفترات السلسلة الزمنية وعددها 30 Lags تقع خارج فترة الثقة (المنطقة المظللة) أي أن جميع هذه الارتباطات ذات أهمية إحصائية ، هل يعني ذلك إمكانية نمذجة السير العشوائي والتنبؤ به؟ الإجابة هي لا. حيث تتطلب معظم خوارزميات التنبؤ بيانات ثابتة Stationary أي ثبات كل من المتوسط ، التباين ، والتغاير (mean, variance, covariance)، والمشي العشوائي لا يعتبر كذلك أي أنه غير ثابت Non-stationary (سنخصص فصلًا كاملا عن مفهوم الثبات Stationarity).

السير العشوائي Random Walk

لن نتعمق في اختبارات الثبات Stationarity والمنطق الكامن وراء هذا المفهوم في هذا الفصل ولكن كقاعدة عامة ، يمكنك جعل السلاسل الزمنية ثابتة عن طريق ما يسمى بحساب فرق الرتبة الأولى first-order difference والذي تم شرحه بالفصل الثاني:

```python
# استيراد المكتبات
import pandas as pd

# First-order difference | احتساب فرق الرتبة الأولى
s_random_walk = pd.Series(random_walk)
s_random_walk_diff = s_random_walk.diff(1).dropna()

# تصوير البيانات
g = sns.lineplot(x=np.arange(len(s_random_walk_diff)), y=s_random_walk_diff,
                 color='#79C9AB')

g.set_title("Random Walk plot First Order Difference", fontsize=20)
g.set_xlabel('Time', fontsize=15)
g.set_ylabel('Value', fontsize=15)
plt.show()
```



التصوير البياني لسلسلة السير العشوائي بعد إجراء الفرق من الرتبة الأولى

```python
s_random_walk_diff.unique()
```

```
array([ 1., -1.])
```

القيم الموجود سلسلة الفرق من الرتبة الأولى هي إما -1 أو 1 و سنشرح ذلك في الشريحة التالية !

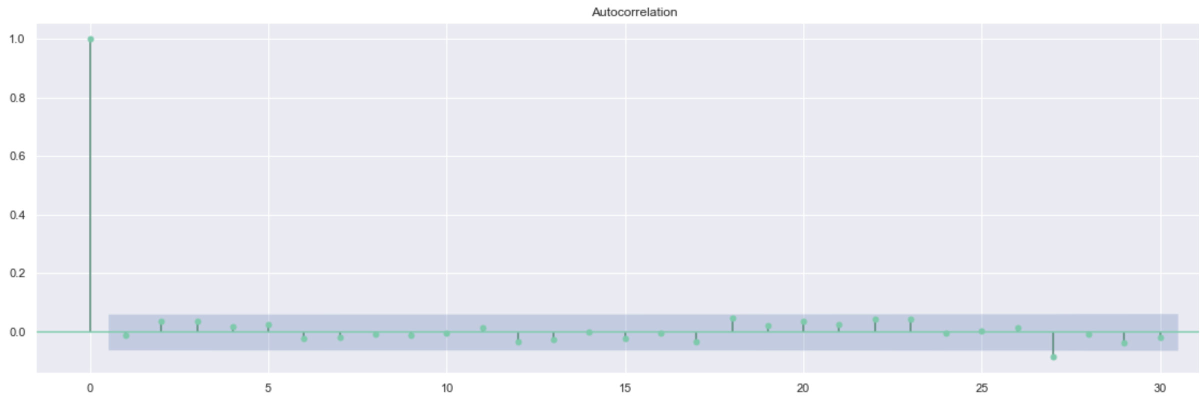
السير العشوائي Random Walk

كما شاهدنا قيم سلسلة السير العشوائي بعد إجراء الفرق من الرتبة الأولى هي -١ أو ١ نظراً لأن الفرق بين القيمة السابقة والحالية هي قيمة الزيادة التي كنا نضيفها في كل فترة زمنية جديدة وهي (-١ أو ١) مثال :

- القيمة السابقة = 0
- القيمة الجديدة = القيمة السابقة + (١ أو -١ حسب التوليد العشوائي) = 0 + ١
- الفرق من الرتبة الأولى = القيمة الجديدة – القيمة السابقة = ١ – 0 = ١

إذا كانت قيم السلسلة تمثل سيراً عشوائيًا فإن المسار العشوائي للفرق من الرتبة الأولى سيبدو كتشويش أبيض. لتأكيد هذا الادعاء ، دعنا نرسم مخطط الارتباط التلقائي لقيم السير العشوائية بعد إجراء الفرق من الرتبة الأولى.

```
sm.graphics.tsa.plot_acf(np.array(s_random_walk_diff), color='#79C9AB',
                         vlines_kwargs= {"colors": '#508672'})
plt.show()
```



كما توقعنا فإن جميع الارتباطات التلقائية للسلسلة لا تمثل أي دلالة إحصائية بالتالي تعبر عن سلسلة تمثل تشويشًا أبيضًا

ملخص Summary

يعد التعرف على مفهوم التشويش الأبيض White Noise و السير العشوائي Random Walk أمرًا ضروريًا لأي مهمة تتضمن السلاسل الزمنية ، فإذا كانت البيانات تبدو مثل التشويش الأبيض أو السير العشوائي ، فلا تهتم بإجراء التنبؤ (توقع السلوك المستقبلي للسلسلة الزمنية)، لأنه لن يصل بك إلى أي مكان. ومع ذلك ، إذا كنت بحاجة ماسة إلى إيجاد تنبؤات ، فقم بتعيين فترات N المتوقعة مساوية للقيمة الأخيرة للسلسلة (سنوضح ذلك فيما بعد) لن تبدو هذه التنبؤات بتلك الجاذبية ولكنها ستقلل من الخطأ نوعًا ما.