How to change to normal distribution

Gaussian-Like و Gaussian

قد تكون هناك مناسبات عندما تعمل بتوزيع غير غاوسي ، لكنك ترغب في استخدام الأساليب الإحصائية البارامترية بدلاً من الأساليب اللامعلمية.

على سبيل المثال ، قد يكون لديك عينة بيانات لها شكل الجرس المألوف ، مما يعني أنها تبدو غاوسية ، لكنها فشلت في واحد أو أكثر من اختبارات الحالة الطبيعية الإحصائية. هذا يشير إلى أن البيانات قد تكون مثل Gaussian. قد تفضل استخدام الإحصائيات البارامترية في هذه الحالة أفضل ولأن البيانات من الواضح أنها غاوسية ، أو يمكن أن تكون ، بعد لقوة إحصائية نظرًا التحويل الصحيح للبيانات.

في هذا المنشور ، سنلقي نظرة على بعض الأساليب البسيطة التي قد تكون قادرًا على استخدامها لتحويل عينة بيانات بتوزيع يشبه Gaussian إلى توزيع Gaussian.

قد تكون هناك حاجة إلى بعض التجريب والحكم. لا يوجد حل سحري لهذه العملية.

حجم العينة

أحد الأسباب الشائعة لكون عينة البيانات غير Gaussian هو أن حجم عينة البيانات صغير جدًا.

ومن ثم ، الحد الأدنى. تم تطوير العديد من الأساليب الإحصائية حيث كانت البيانات شحيحة قد يصل عدد العينات للعديد من الطرق إلى 20 أو 30 ملاحظة.

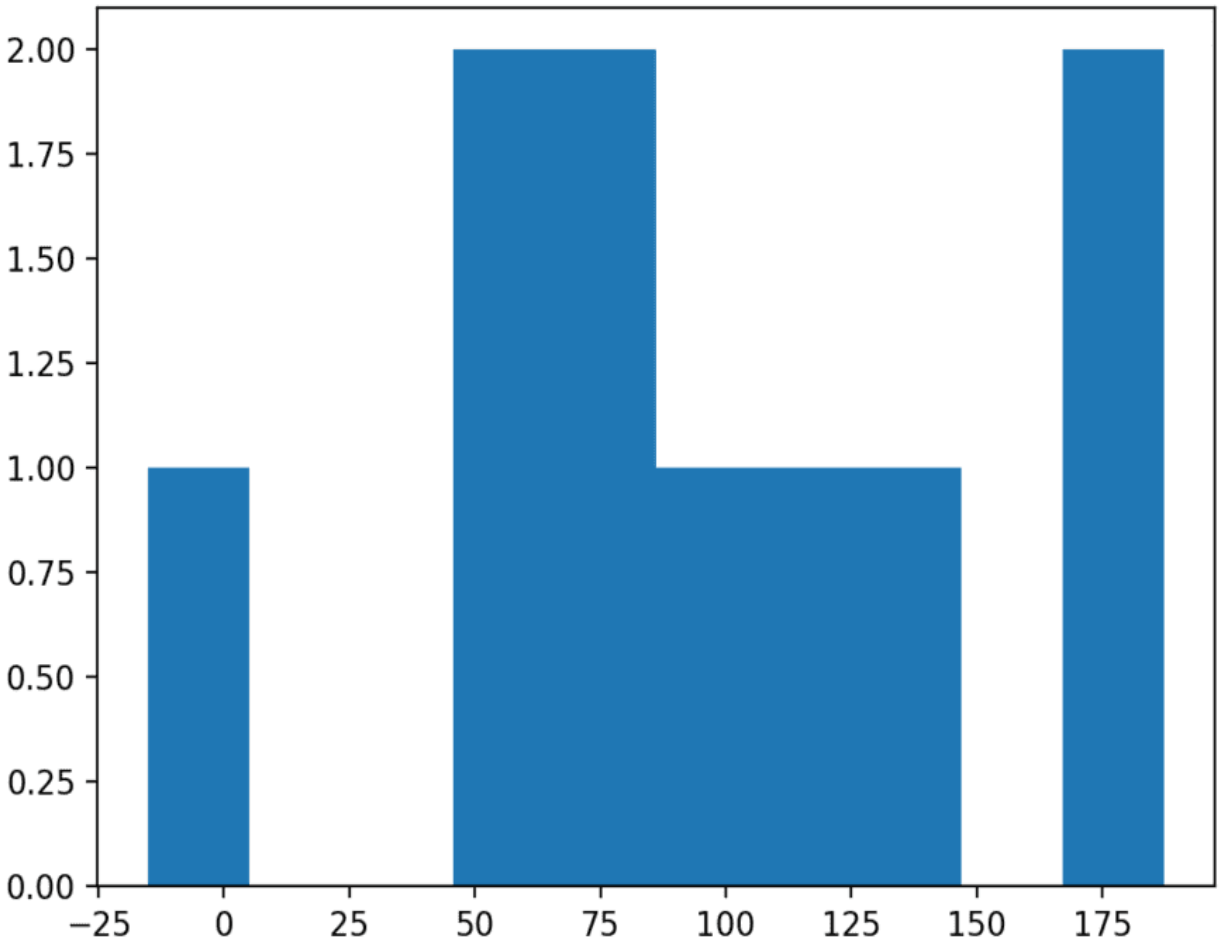
ومع ذلك ، نظرًا للضوضاء في بياناتك ، فقد لا ترى شكل الجرس المألوف أو تفشل في اختبارات الحالة الطبيعية مع عدد متواضع من العينات ، مثل 50 أو 100. إذا كان هذا هو الحال ، فربما بفضل قانون الأعداد الكبيرة ، كلما زادت البيانات التي تجمعها ، يمكنك جمع المزيد من البيانات زادت احتمالية استخدام بياناتك لوصف التوزيع الأساسي للسكان.

لجعل هذا ملموسًا ، فيما يلي مثال على مخطط لعينة صغيرة من 50 ملاحظة مأخوذة من توزيع غاوسي بمتوسط 100 وانحراف معياري قدره 50.

```
1  # histogram plot of a small sample
2  from numpy.random import seed
```

```
3  from numpy.random import randn
4  from matplotlib import pyplot
5  # seed the random number generator
6  seed(1)
7  # generate a univariate data sample
8  data = 50 * randn(50) + 100
9  # histogram
10 pyplot.hist(data)
11 pyplot.show()
```

يؤدي تشغيل المثال إلى إنشاء رسم بياني للبيانات يظهر عدم وجود توزيع غاوسي واضح ، ولا حتى توزيع غاوسي.



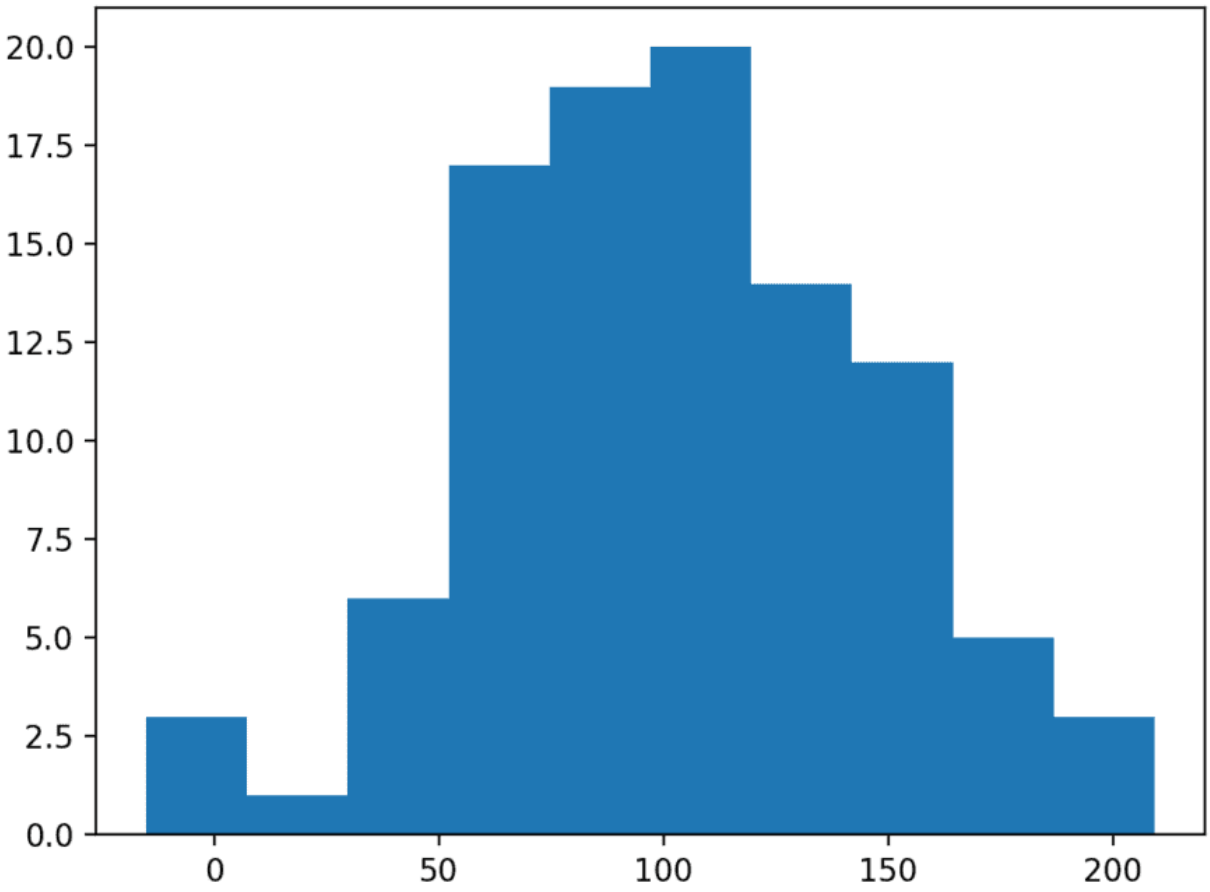
رسم بياني لعينة بيانات صغيرة جدًا

يمكن أن تساعد زيادة حجم العينة من 50 إلى 100 في عرض الشكل الغاوسي لتوزيع البيانات بشكل أفضل.

```
1  # histogram plot of a small sample
2  from numpy.random import seed
3  from numpy.random import randn
4  from matplotlib import pyplot
```

```
5  # seed the random number generator
6  seed(1)
7  # generate a univariate data sample
8  data = 50 * randn(100) + 100
9  # histogram
10 pyplot.hist(data)
11 pyplot.show()
```

باستخدام هذا المثال ، يمكننا أن نرى بشكل أفضل التوزيع الغاوسي للبيانات التي ستجتاز كلاً من الاختبارات الإحصائية وفحوصات كرة العين.



رسم بياني لعينة بيانات أكبر

دقة البيانات

ربما تتوقع توزيعًا غاوسيًا من البيانات ، ولكن بغض النظر عن حجم العينة التي تجمعها ، فإنها لا تتحقق.

قد يتم حجب توزيع البيانات من .السبب الشائع لذلك هو الدقة التي تستخدمها لتجميع الملاحظات قد يكون هناك العديد من الأسباب لتعديل دقة .خلال الدقة المختارة للبيانات أو دقة الملاحظات البيانات قبل النمذجة ، مثل:

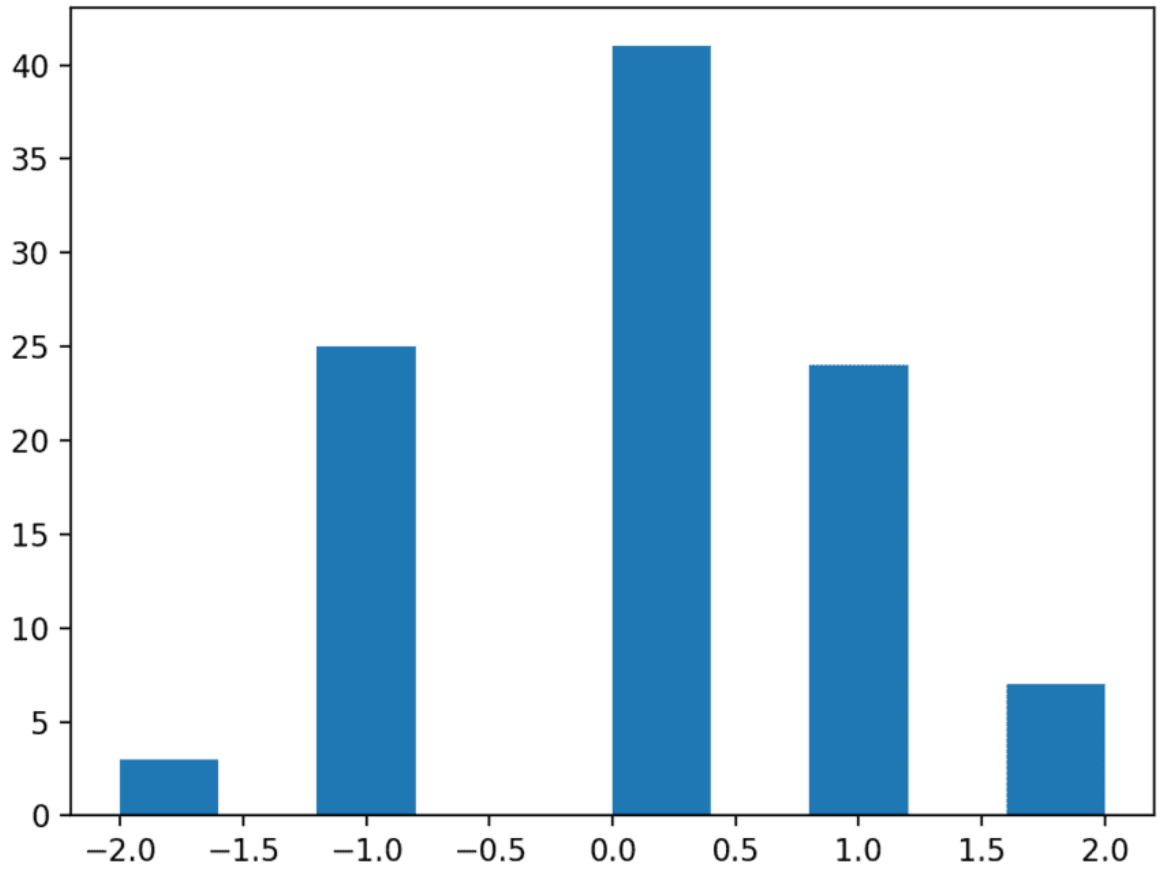
- تكوين آلية عمل الملاحظة.
- البيانات تمر عبر عملية مراقبة الجودة.

- دقة قاعدة البيانات المستخدمة لتخزين البيانات.

بمتوسط 0 وانحراف رقم غاوسي عشوائي لجعل هذا الأمر ملموسًا ، يمكننا عمل عينة من 100 معياري قدره 1 وإزالة جميع المنازل العشرية.

```
# histogram plot of a low res sample
from numpy.random import seed
from numpy.random import randn
from matplotlib import pyplot
# seed the random number generator
seed(1)
# generate a univariate data sample
data = randn(100)
# remove decimal component
data = data.round(0)
# histogram
pyplot.hist(data)
pyplot.show()
```

ستؤدي إضافة Gaussian. تشغيل المثال ينتج عنه توزيع يبدو منفصلاً على الرغم من أنه يشبه الدقة إلى الملاحظات إلى توزيع أكثر اكتمالاً للبيانات.



رسم بياني لعينة بيانات منخفضة الدقة

القيم المتطرفة

قد يكون لعينة البيانات توزيع غاوسي ، ولكن قد يتم تشويهها لعدد من الأسباب.

يمكن أن توجد القيم المتطرفة لعدد من .السبب الشائع هو وجود قيم متطرفة عند حافة التوزيع الأسباب ، مثل:

- خطأ في القياس.
- بيانات مفقودة.
- تلف البيانات.
- أحداث نادرة.

في مثل هذه الحالات ، يمكن تحديد القيم القصوى وإزالتها من أجل جعل التوزيع أكثر
.غالبًا ما تسمى هذه القيم المتطرفة القيم المتطرفة .غاوسيًا

قد يتطلب ذلك خبرة في المجال أو استشارة أحد خبراء المجال من أجل تصميم معايير لتحديد القيم المتطرفة ثم إزالتها من عينة البيانات وجميع عينات البيانات التي تتوقع أنت أو نموذجك العمل
.معها في المستقبل

.يمكننا توضيح مدى سهولة وجود قيم متطرفة تعطل توزيع البيانات

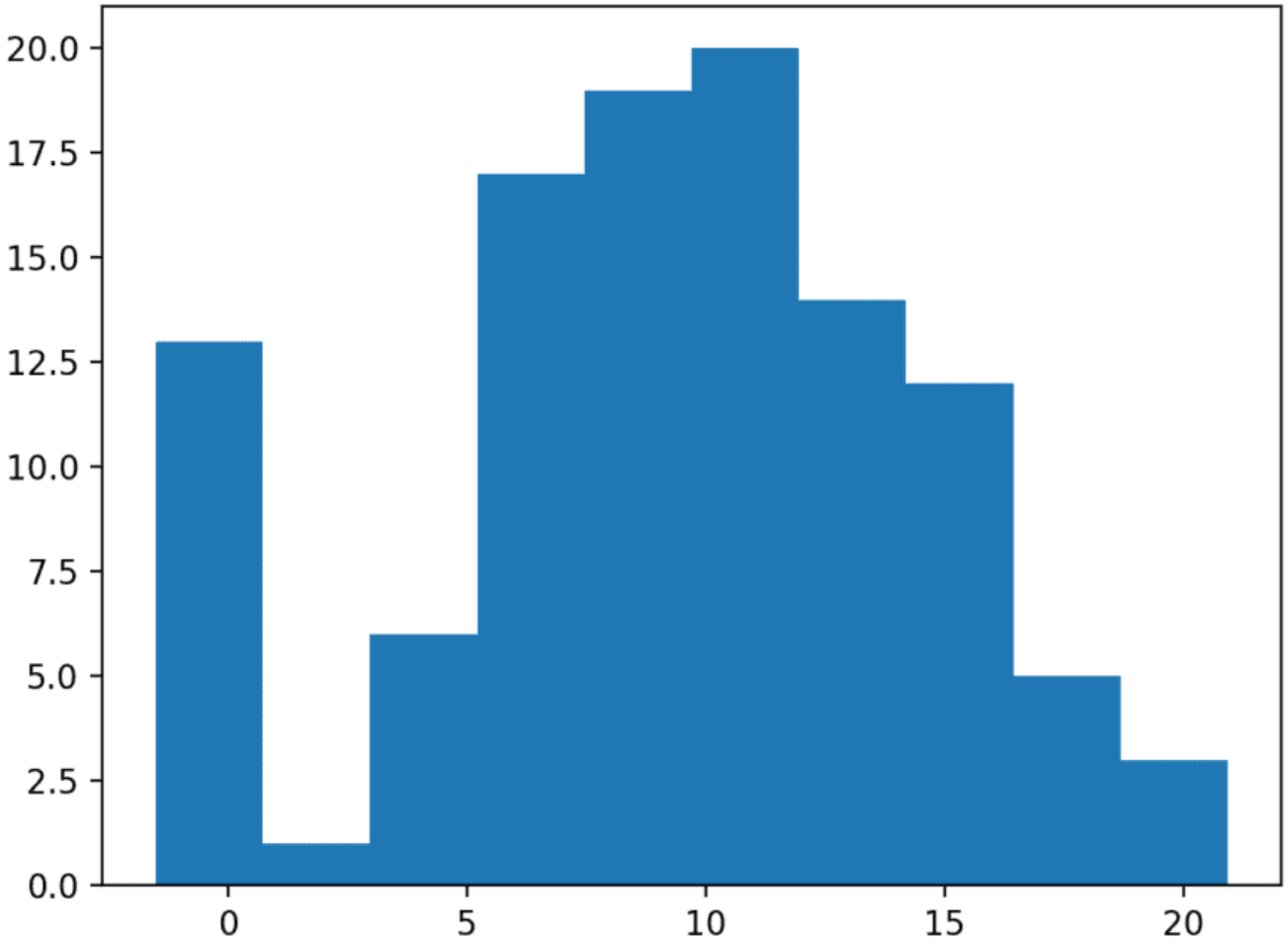
يُنشئ المثال أدناه عينة بيانات تحتوي على 100 رقم غاوسي عشوائي تم قياسه بمتوسط 10 وانحراف معياري قدره 5. ثم تتم إضافة 10 ملاحظات إضافية ذات قيمة صفرية إلى
هذا سلوك شائع .يمكن أن يحدث هذا إذا تم تعيين القيم المفقودة أو التالفة بقيمة صفر .التوزيع
.فمثلا في مجموعات بيانات التعلم الآلي المتاحة للجمهور ؛

```
# histogram plot of data with outliers
from numpy.random import seed
from numpy.random import randn
from numpy import zeros
from numpy import append
from matplotlib import pyplot
# seed the random number generator
seed(1)
# generate a univariate data sample
data = 5 * randn(100) + 10
# add extreme values
data = append(data, zeros(10))
# histogram
```

14 pyplot.hist(data)

15 pyplot.show()

يمكنك أن ترى بوضوح كيف أن التردد العالي .يؤدي تشغيل المثال إلى إنشاء ورسم عينة البيانات
.غير المتوقع للملاحظات ذات القيمة الصفرية يعطل التوزيع



رسم بياني لعينة البيانات ذات القيم القصوى

ذيول طويلة

بالإضافة إلى وفرة الأحداث النادرة على حافة التوزيع ، .يمكن أن تظهر القيم المتطرفة بعدة طرق .قد ترى ذيلًا طويلاً للتوزيع في أحد الاتجاهين أو كلاهما

في المؤامرات ، يمكن أن يجعل هذا التوزيع يبدو وكأنه أسي ، بينما في الواقع قد يكون غاوسيًا .مع وفرة من الأحداث النادرة في اتجاه واحد

يمكنك استخدام قيم حد بسيطة ، ربما بناءً على عدد الانحرافات المعيارية عن المتوسط ، لتحديد .قيم الذيل الطويل وإزالتها

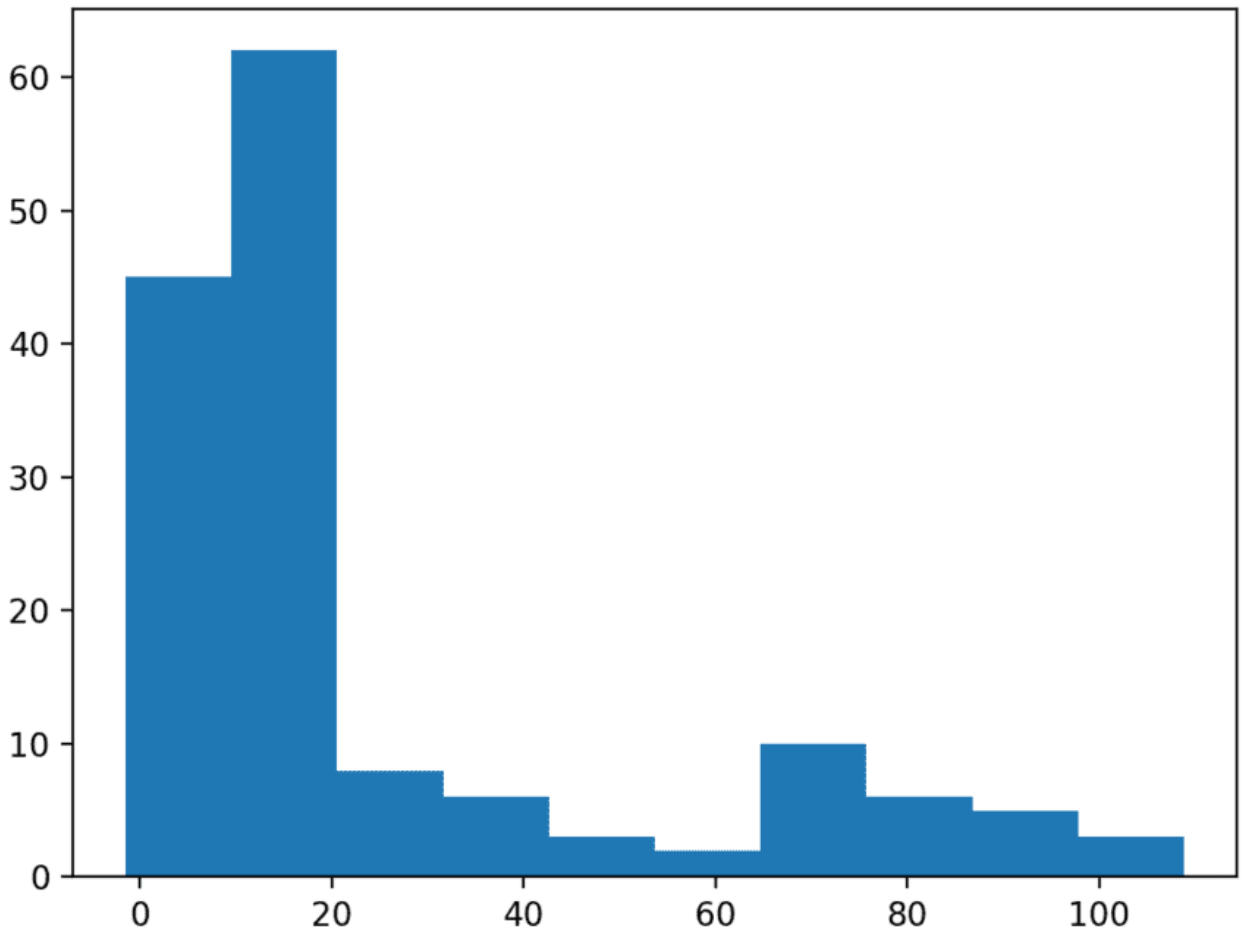
يمكننا إثبات ذلك بمثال مفتعل. تحتوي عينة البيانات على 100 رقم عشوائي غاوسي بمتوسط 10 وانحراف معياري 5. تمت إضافة 50 قيمة عشوائية منتظمة في النطاق من 10 إلى 110. هذا .يخلق ذيل طويل على التوزيع

```
1  # histogram plot of data with a long tail
2  from numpy.random import seed
3  from numpy.random import randn
4  from numpy.random import rand
5  from numpy import append
6  from matplotlib import pyplot
7  # seed the random number generator
8  seed(1)
9  # generate a univariate data sample
10 data = 5 * randn(100) + 10
11 tail = 10 + (rand(50) * 100)
12 # add long tail
13 data = append(data, tail)
14 # histogram
```

15 pyplot.hist(data)

16 pyplot.show()

عند تشغيل المثال ، يمكنك أن ترى كيف يشوه الذيل الطويل التوزيع الغاوسي ويجعله يبدو أسيًا (تقريبًا أو ربما حتى ثنائي النسق (نتوءان.



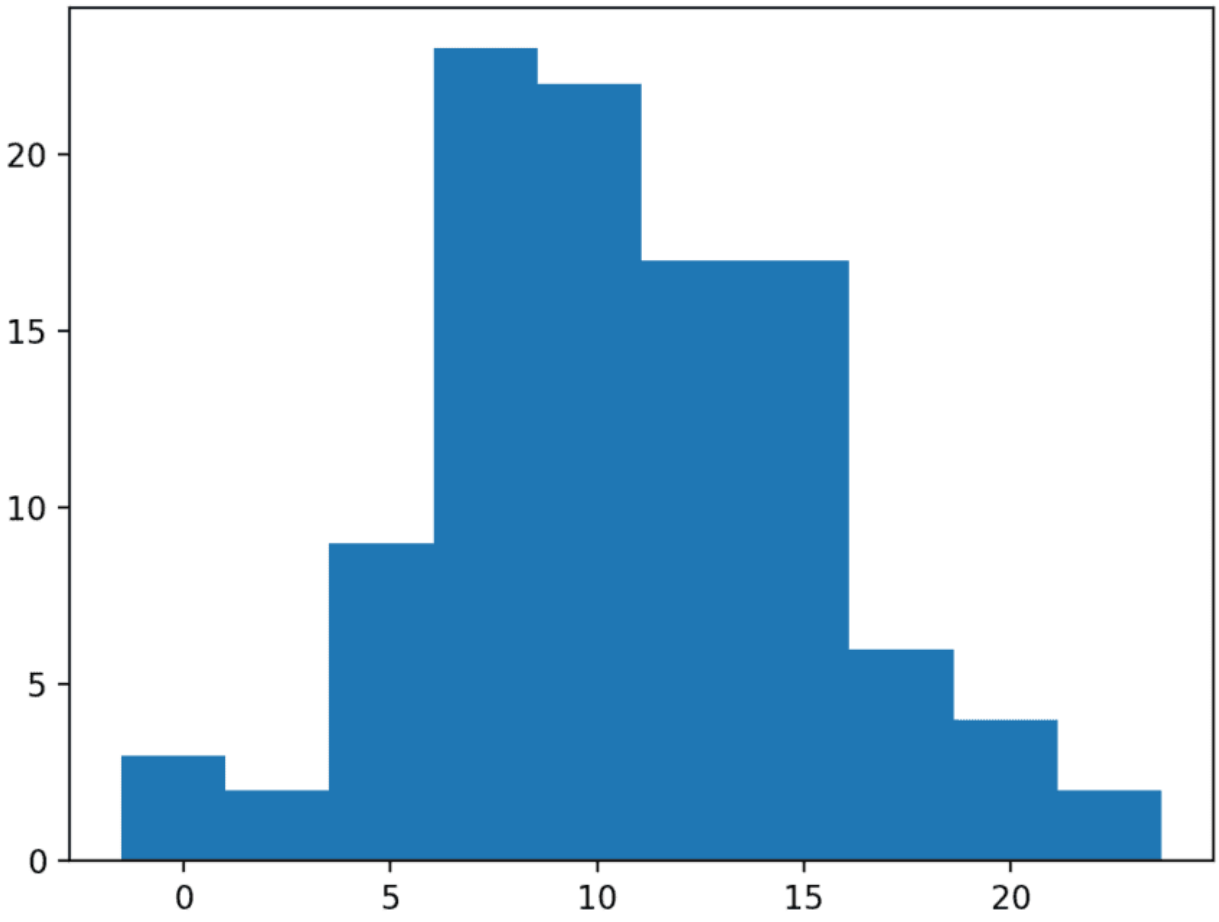
رسم بياني لعينة البيانات ذات الذيل الطويل

يمكننا استخدام حد بسيط ، مثل القيمة 25 ، في مجموعة البيانات هذه كقطع وإزالة جميع الملاحظات الأعلى من هذا الحد. لقد اخترنا هذا الحد بمعرفة مسبقة عن كيفية إنشاء عينة

البيانات ، ولكن يمكنك تخيل اختبار عتبات مختلفة على مجموعة البيانات الخاصة بك وتقييم تأثيرها.

```
# histogram plot of data with a long tail
from numpy.random import seed
from numpy.random import randn
from numpy.random import rand
from numpy import append
from matplotlib import pyplot
# seed the random number generator
seed(1)
# generate a univariate data sample
data = 5 * randn(100) + 10
tail = 10 + (rand(10) * 100)
# add long tail
data = append(data, tail)
# trim values
data = [x for x in data if x < 25]
# histogram
pyplot.hist(data)
pyplot.show()
```

يُظهر تشغيل الكود كيف يؤدي هذا القص البسيط للذيل الطويل إلى إرجاع البيانات إلى توزيع غاوسي.



رسم بياني لعينة البيانات مع ذيل طويل مبتور

تحويلات الطاقة

قد يكون توزيع البيانات طبيعيًا ، ولكن قد تتطلب البيانات تحويلًا للمساعدة في كشفها.

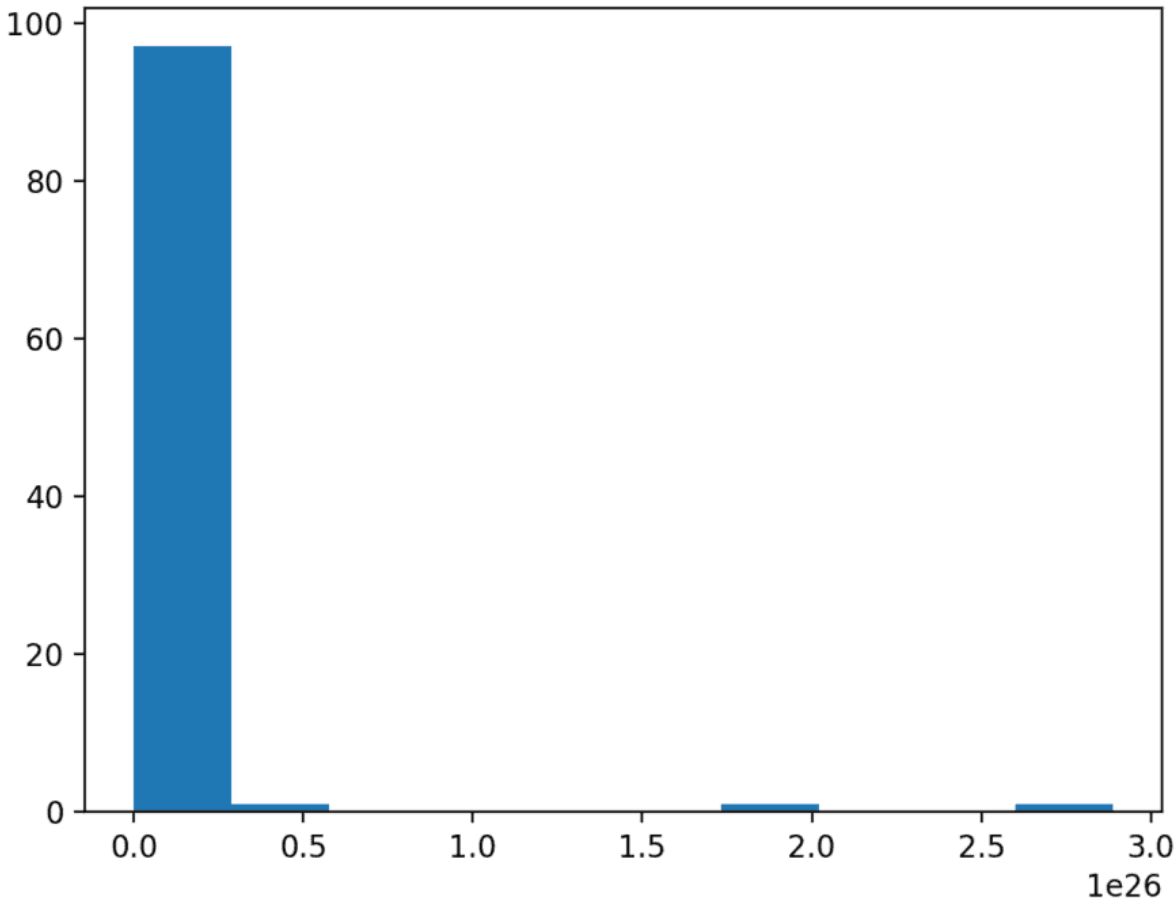
على سبيل المثال ، قد تحتوي البيانات على انحراف ، مما يعني أن الجرس في شكل الجرس قد في بعض الحالات ، يمكن تصحيح ذلك عن طريق تحويل البيانات عن .يتم دفعه بطريقة أو بأخرى طريق حساب الجذر التربيعي للملاحظات.

بالتناوب ، قد يكون التوزيع أسيًا ، ولكن قد يبدو طبيعيًا إذا تم تحويل الملاحظات عن طريق أخذ تسمى البيانات مع هذا التوزيع السجل العادي .اللوغاريتم الطبيعي للقيم.

لجعل هذا ملموسًا ، يوجد أدناه مثال لعينة من الأرقام الغوسية التي تم تحويلها ليكون لها توزيع أسي.

```
# log-normal distribution
from numpy.random import seed
from numpy.random import randn
from numpy import exp
from matplotlib import pyplot
# seed the random number generator
seed(1)
# generate two sets of univariate observations
data = 5 * randn(100) + 50
# transform to be exponential
data = exp(data)
# histogram
pyplot.hist(data)
pyplot.show()
```

ليس من الواضح أن البيانات يؤدي تشغيل المثال إلى إنشاء رسم بياني يوضح التوزيع الأسي. هي في الواقع لوغاريتمي عادي.



الرسم البياني للتوزيع العادي للسجل

أخذ الجذر التربيعي ولوغاريتم الملاحظة من أجل جعل التوزيع طبيعيًا ينتمي إلى فئة من هي طريقة لتحويل البيانات قادرة على أداء Box-Cox طريقة .التحويلات تسمى تحويلات الطاقة تم تسمية هذه الطريقة باسم .مجموعة من تحويلات الطاقة ، بما في ذلك السجل والجذر التربيعي .جورج بوكس وديفيد كوكس

يمكن .أكثر من ذلك ، يمكن تهيئته لتقييم مجموعة من التحويلات تلقائيًا واختيار أفضل ملاءمة قد تكون عينة .اعتباره أداة قوية لتسوية التغيير المعتمد على الطاقة في عينة البيانات الخاصة بك البيانات الناتجة أكثر خطية وستمثل بشكل أفضل التوزيع الأساسي غير للطاقة ، بما في ذلك Gaussian.

، تتحكم lambda يتطلب الأمر حجة ، تسمى .Box-Cox طريقة boxcox () SciPy وظيفة تنفذ .في نوع التحويل المطلوب إجراؤه

فيما يلي بعض القيم الشائعة لـ lambda:

- هو تحويل متبادل . لامدا = -1.
- تحويل جذري تربيعي متبادل لامدا = -0.5.
- تحويل سجل لامدا = 0.0.
- تحويل جذر تربيعي لامدا = 0.5.
- لا يوجد تحويل لامدا = 1.0.

لإجراء Box-Cox على سبيل المثال ، نظرًا لأننا نعلم أن البيانات غير طبيعية ، يمكننا استخدام تحويل السجل عن طريق تعيين lambda 0 صراحةً على.

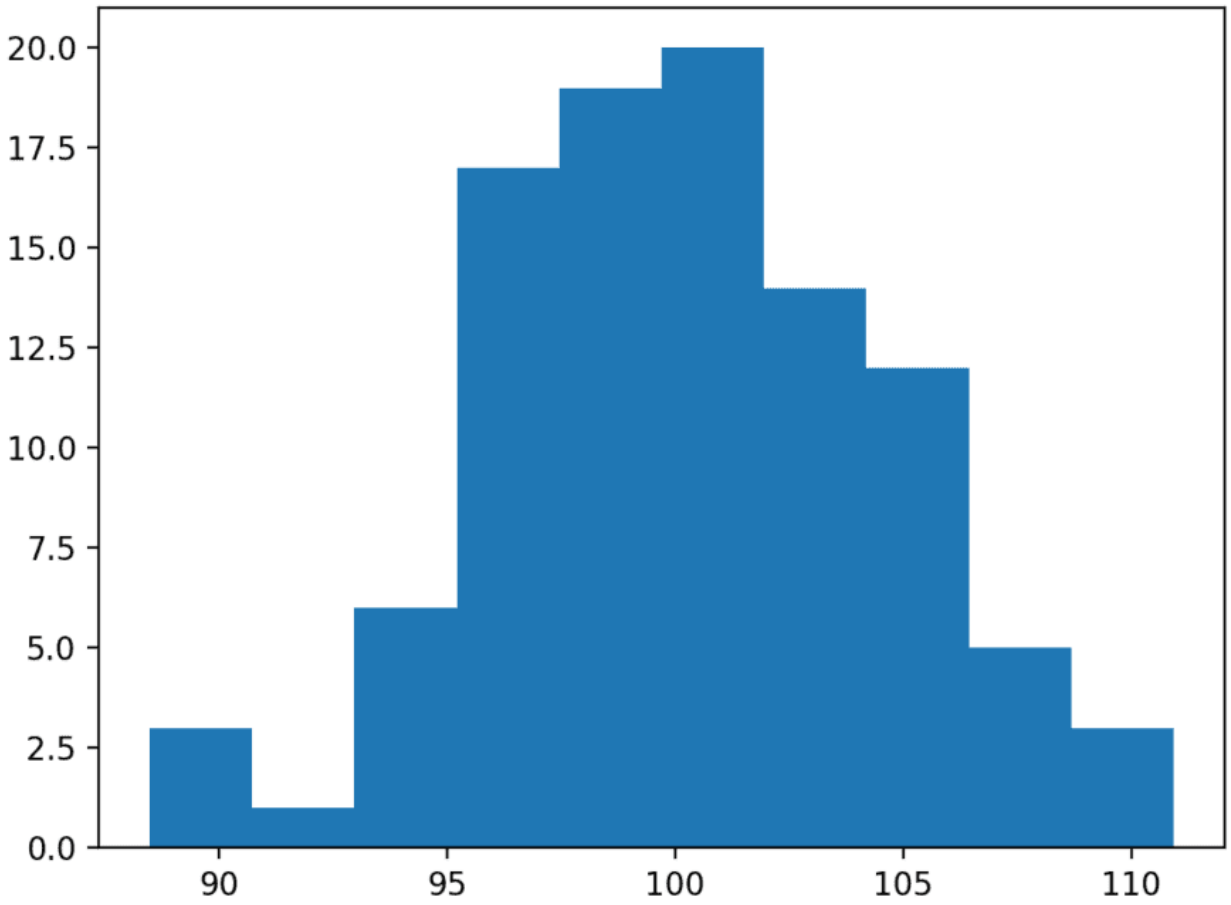
```
# power transform
data = boxcox(data, 0)
```

المثال الكامل لتطبيق تحويل Box-Cox أدناه مُدرج على عينة البيانات الأسية.

```
# box-cox transform
from numpy.random import seed
from numpy.random import randn
from numpy import exp
from scipy.stats import boxcox
from matplotlib import pyplot
# seed the random number generator
seed(1)
# generate two sets of univariate observations
data = 5 * randn(100) + 100
```

```
11 # transform to be exponential
12 data = exp(data)
13 # power transform
14 data = boxcox(data, 0)
15 # histogram
16 pyplot.hist(data)
17 pyplot.show()
```

على عينة البيانات ورسم النتيجة ، مع إظهار Box-Cox يؤدي تشغيل المثال إلى إجراء تحويل التوزيع الغاوسي بوضوح.



مخطط الرسم البياني لمربع كوكس عينة البيانات الأسية المحولة

.في أنه يفترض أن جميع القيم في عينة البيانات موجبة Box-Cox يتمثل أحد قيود تحويل

Yeo-Johnson. طريقة بديلة لا تجعل هذا الافتراض هو تحول