**violinplot**

Boxplot ve KDE plot ın bir kombinasyonunu çizer.

Violinplot, boxplot ve whiskerplot ile benzer bir rol oynar. Bir veya daha fazla kategorik değişkenin çeşitli seviyelerinde nicel verilerin dağılımını göstererek bu dağılımların karşılaştırılmasına imkan sağlar. Tüm çizim bileşenlerinin gerçek veri noktalarına karşılık geldiği bir boxplot tan farklı olarak, violinplot, temel dağılımın bir çekirdek yoğunluğu tahminini (KDE) içerir.

Aynı anda birden fazla veri dağılımını göstermenin etkili bir yolu olmasına rağmen tahmin prosedürünün numune boyutundan etkilendiğini ve nispeten küçük numuneler için violinplot ın yanıltıcı bir şekilde pürüzsüz görünebileceği unutulmamalıdır.

Veriler, aşağıda gösterilenler de dahil olmak üzere çeşitli biçimlerde iletilebilir:

* Vectors of data represented as lists, numpy arrays, or pandas Series objects passed directly to the x, y, and/or hue parameters.
* A “long-form” DataFrame, in which case the x, y, and hue variables will determine how the data are plotted.
* A “wide-form” DataFrame, such that each numeric column will be plotted.
* An array or list of vectors.

Çoğu durumda, numpy veya Python nesneleri kullanmak mümkün olmasına rağmen, ilişkili adlar eksenlere açıklama eklemek için kullanılacağından pandas nesneleri tercih edilir. İlave olarak, çizim öğelerinin sırasını kontrol etmek maksadıyla gruplama değişkenleri için Kategorik türler kullanılabilir.

Bu işlev, değişkenlerden birini her zaman kategorik olarak ele alır ve veriler sayısal veya tarih türünde olsa bile ilgili eksende (0, 1, … n) sıra konumlarında gösterir.

**Parameters**

**x, y, hue*names of variables in data or vector data, optional***

Inputs for plotting long-form data. See examples for interpretation.

**data*DataFrame, array, or list of arrays, optional***

Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.

**order, hue\_order*lists of strings, optional***

Order to plot the categorical levels in, otherwise the levels are inferred from the data objects.

**bw*{‘scott’, ‘silverman’, float}, optional***

Either the name of a reference rule or the scale factor to use when computing the kernel bandwidth. The actual kernel size will be determined by multiplying the scale factor by the standard deviation of the data within each bin.

**cut*float, optional***

Distance, in units of bandwidth size, to extend the density past the extreme datapoints. Set to 0 to limit the violin range within the range of the observed data (i.e., to have the same effect as trim=True in ggplot.

**scale*{“area”, “count”, “width”}, optional***

The method used to scale the width of each violin. If area, each violin will have the same area. If count, the width of the violins will be scaled by the number of observations in that bin. If width, each violin will have the same width.

**scale\_hue*bool, optional***

When nesting violins using a hue variable, this parameter determines whether the scaling is computed within each level of the major grouping variable (scale\_hue=True) or across all the violins on the plot (scale\_hue=False).

**gridsize*int, optional***

Number of points in the discrete grid used to compute the kernel density estimate.

**width*float, optional***

Width of a full element when not using hue nesting, or width of all the elements for one level of the major grouping variable.

**inner*{“box”, “quartile”, “point”, “stick”, None}, optional***

Representation of the datapoints in the violin interior. If box, draw a miniature boxplot. If quartiles, draw the quartiles of the distribution. If point or stick, show each underlying datapoint. Using None will draw unadorned violins.

**split*bool, optional***

When using hue nesting with a variable that takes two levels, setting split to True will draw half of a violin for each level. This can make it easier to directly compare the distributions.

**dodge*bool, optional***

When hue nesting is used, whether elements should be shifted along the categorical axis.

**orient*“v” | “h”, optional***

Orientation of the plot (vertical or horizontal). This is usually inferred based on the type of the input variables, but it can be used to resolve ambiguitiy when both x and y are numeric or when plotting wide-form data.

**linewidth*float, optional***

Width of the gray lines that frame the plot elements.

**color*matplotlib color, optional***

Color for all of the elements, or seed for a gradient palette.

**palette*palette name, list, or dict***

Colors to use for the different levels of the hue variable. Should be something that can be interpreted by **[color\_palette()](https://seaborn.pydata.org/generated/seaborn.color_palette.html" \l "seaborn.color_palette" \o "seaborn.color_palette)**, or a dictionary mapping hue levels to matplotlib colors.

**saturation*float, optional***

Proportion of the original saturation to draw colors at. Large patches often look better with slightly desaturated colors, but set this to 1 if you want the plot colors to perfectly match the input color spec.

**ax*matplotlib Axes, optional***

Axes object to draw the plot onto, otherwise uses the current Axes.









