

Applied Data Analytics

Visualisations with plotly

Modifying plotly graphs

Hans-Martin von Gaudecker and Aapo Stenhammar

Motivation

Clearly communicating what is on your graph is absolutely crucial for readers to understand what is in it!

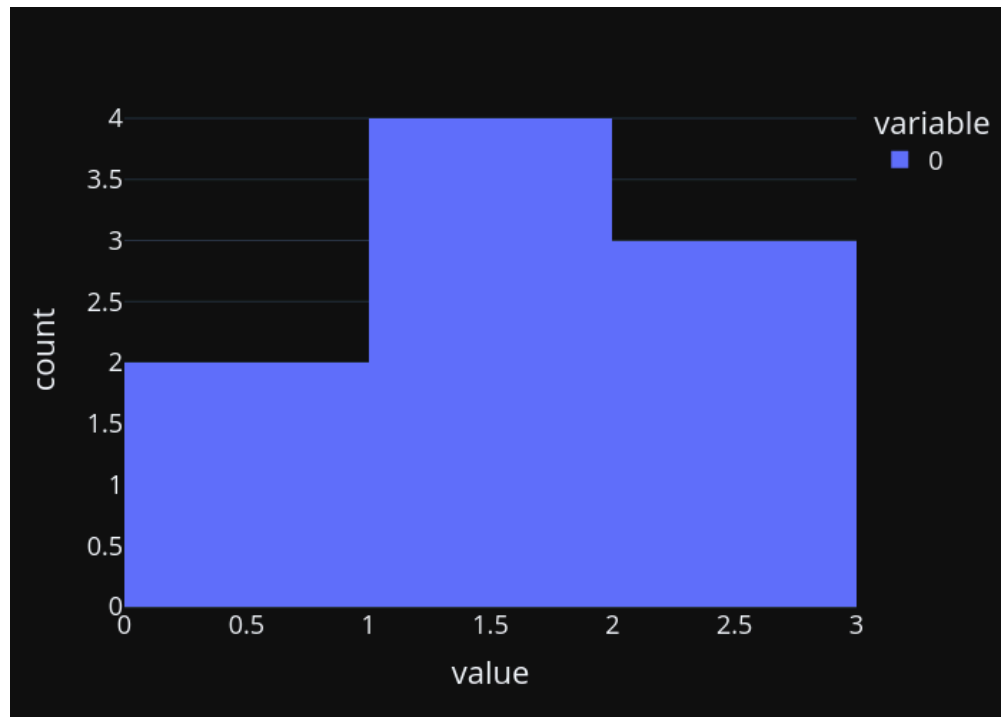
- Title
- Axis labels, including units (!!!)
- *(Legends)*
- Bonus: Additional features, like vertical lines

Example series

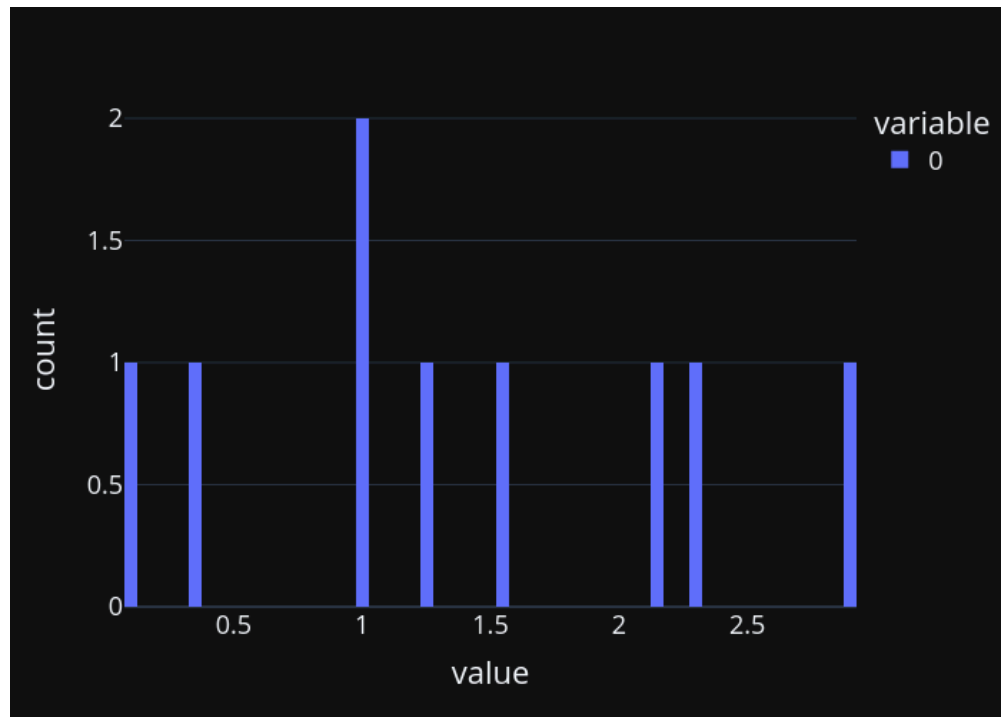
```
[1] monthly_income = pd.Series(  
    [  
        1.57,  
        0.09,  
        1,  
        2.9,  
        1.25,  
        1,  
        0.35,  
        2.3,  
        2.15  
    ],  
)
```

```
[2] monthly_income  
[2] 0    1.57  
    1    0.09  
    2    1.00  
    3    2.90  
    4    1.25  
    5    1.00  
    6    0.35  
    7    2.30  
    8    2.15  
dtype: float64
```

```
monthly_income.plot.hist()
```



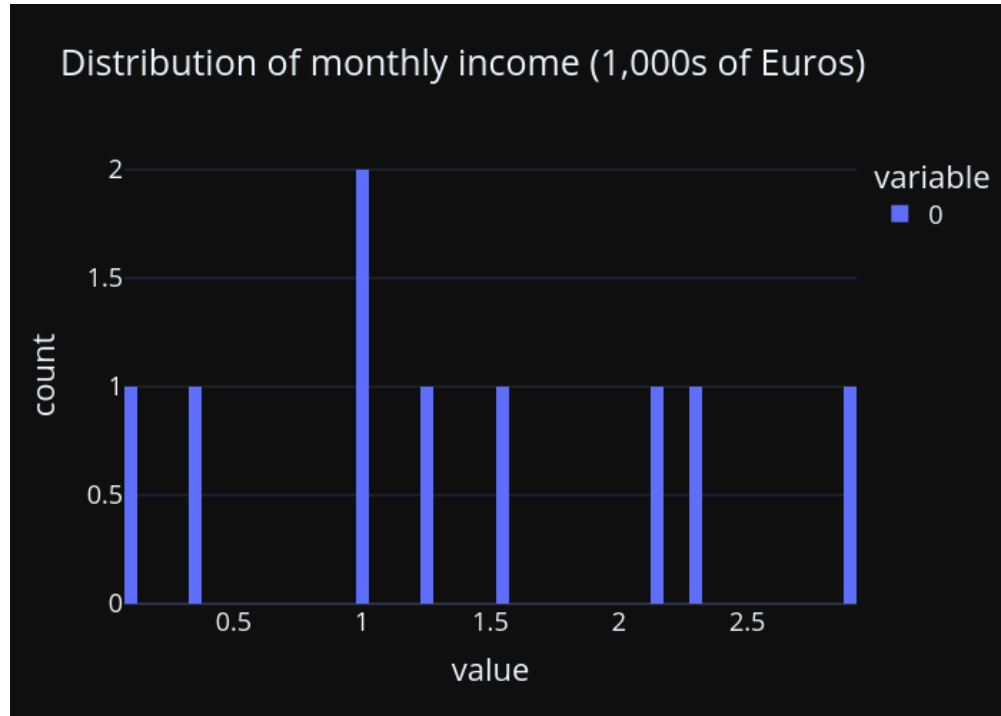
```
monthly_income.plot.hist(nbins=100)
```



Adding a title

```
monthly_income.plot.hist(  
    nbins=100,  
    title="Distribution of monthly income (1,000s of Euros)",  
)
```

Adding a title

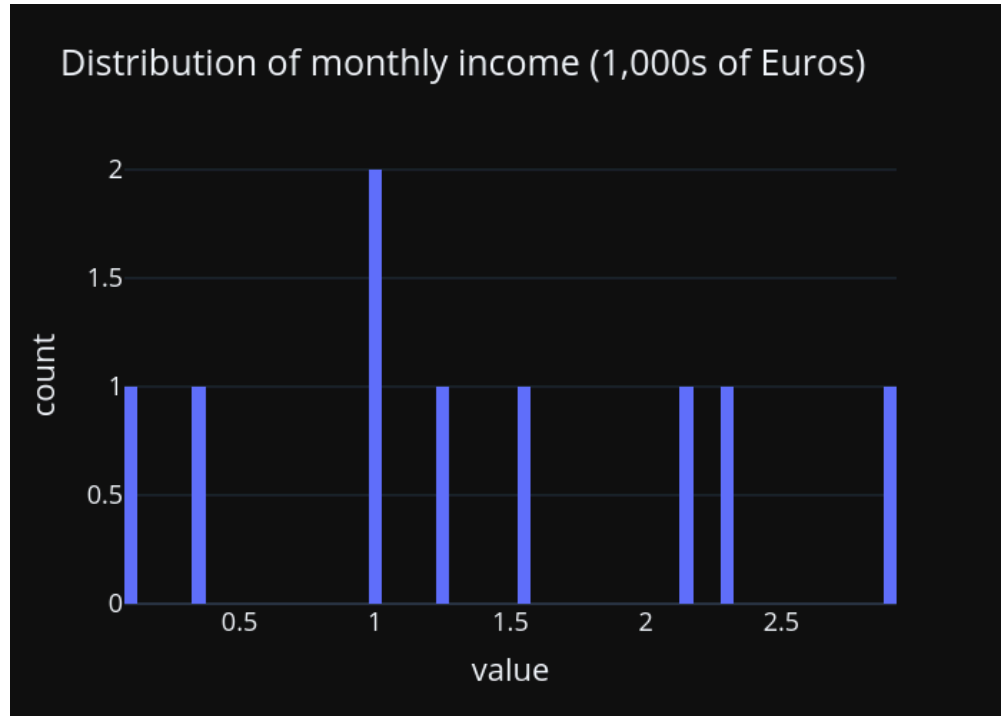


Removing the legend

```
fig = monthly_income.plot.hist(  
    nbins=100,  
    title="Distribution of monthly income (1,000s of Euros)",  
)  
fig.update_layout(showlegend=False)  
fig
```

1. Assign the figure to a new variable
2. Call the `.update_layout()` method

Removing the legend

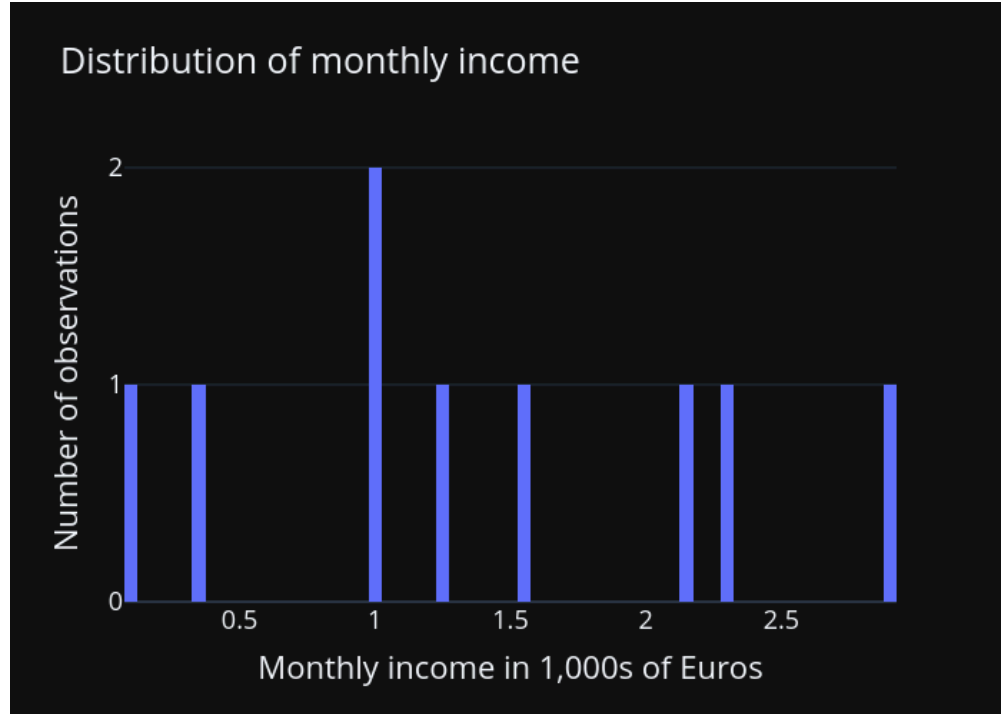


Making the axes nice

```
fig = monthly_income.plot.hist(  
    nbins=100,  
    title="Distribution of monthly income",  
)  
fig.update_layout(  
    showlegend=False,  
    xaxis_title="Monthly income in 1,000s of Euros",  
    yaxis_title="Number of observations",  
    yaxis_tickvals=[0, 1, 2],  
)  
fig
```

1. Sensible titles for x- and y-axes
2. Sensible ticks for y-axis

Making the axes nice



Adding the mean

```
fig = monthly_income.plot.hist(  
    nbins=100,  
    title="Distribution of monthly income",  
)  
fig.update_layout(  
    showlegend=False,  
    xaxis_title="Monthly income in 1,000s of Euros"  
    yaxis_title="Number of observations",  
    yaxis_tickvals=[0, 1, 2],  
)  
fig.add_vline(x=monthly_income.mean(), line_width=3, line_color="red")  
fig
```

Call the `.add_vline()` method on the graph object

Adding the mean

