Applied Data Analytics

Visualisations with plotly

Logarithmic scales

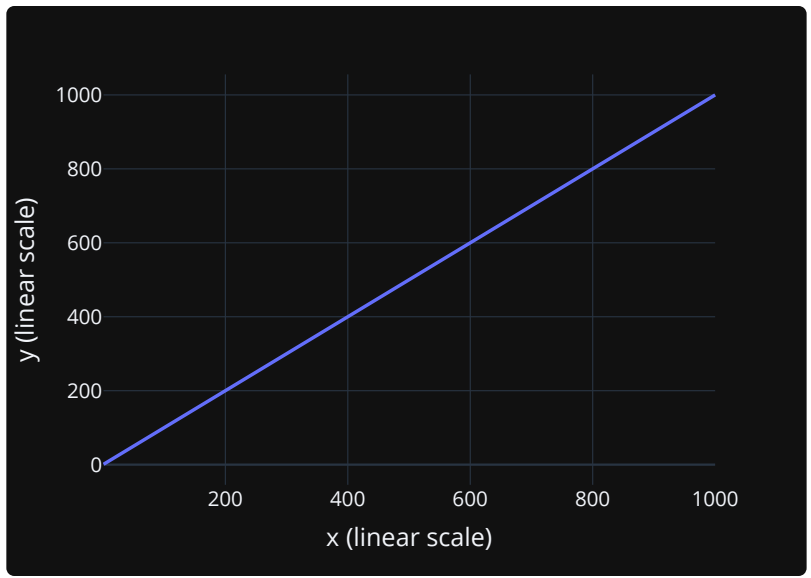
Hans-Martin von Gaudecker

Example data

x	y
1	1
2	2
3	3
⋮	⋮
999	999
1000	1000

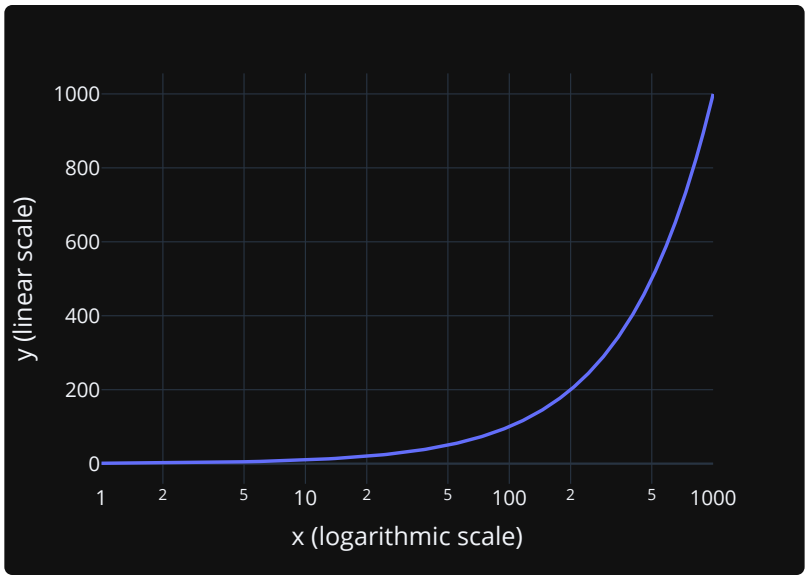
linear x, linear y scales

```
fig = series.plot.line()
fig.update_layout(
    showlegend=False,
    xaxis_title="x (linear scale)",
    yaxis_title="y (linear scale)",
)
fig.show()
```



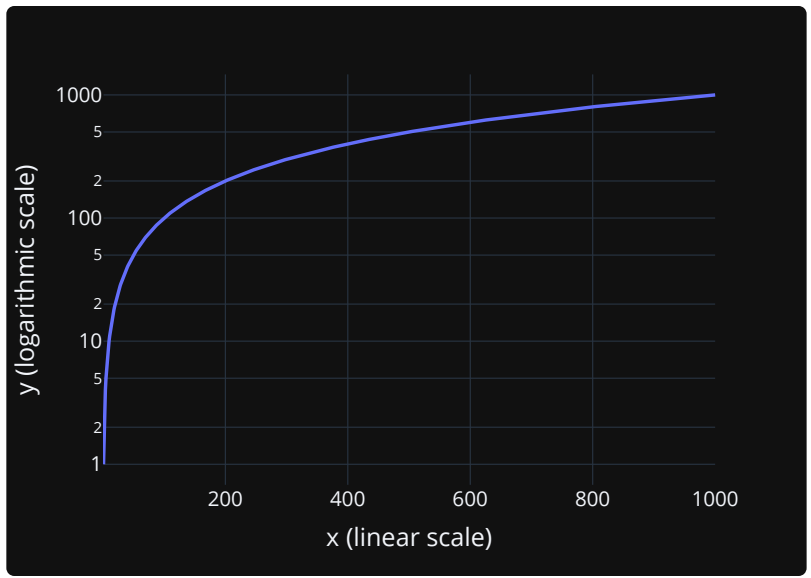
logarithmic x, linear y scales

```
fig = series.plot.line()
fig.update_layout(
    showlegend=False,
    xaxis_type="log",
    xaxis_title="x (logarithmic scale)",
    yaxis_title="y (linear scale)",
)
fig.show()
```



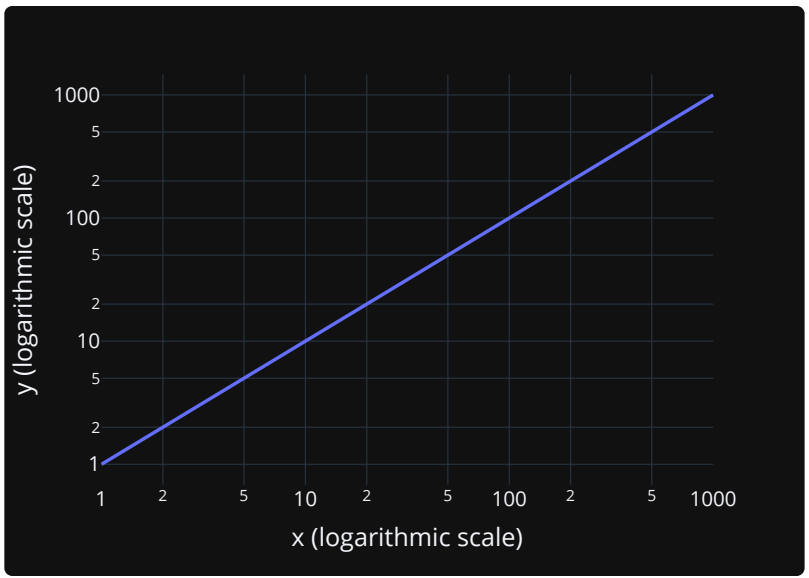
linear x, logarithmic y scales

```
fig = series.plot.line()
fig.update_layout(
    showlegend=False,
    xaxis_type="linear",
    yaxis_type="log",
    xaxis_title="x (linear scale)",
    yaxis_title="y (logarithmic scale)",
)
fig.show()
```



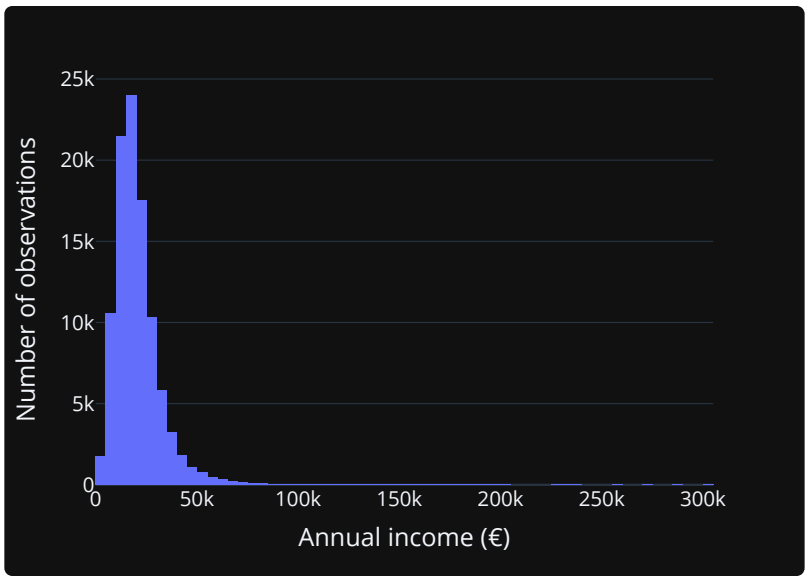
logarithmic x, logarithmic y scales

```
fig = series.plot.line()
fig.update_layout(
    showlegend=False,
    xaxis_type="log",
    yaxis_type="log",
    xaxis_title="x (logarithmic scale)",
    yaxis_title="y (logarithmic scale)",
)
fig.show()
```



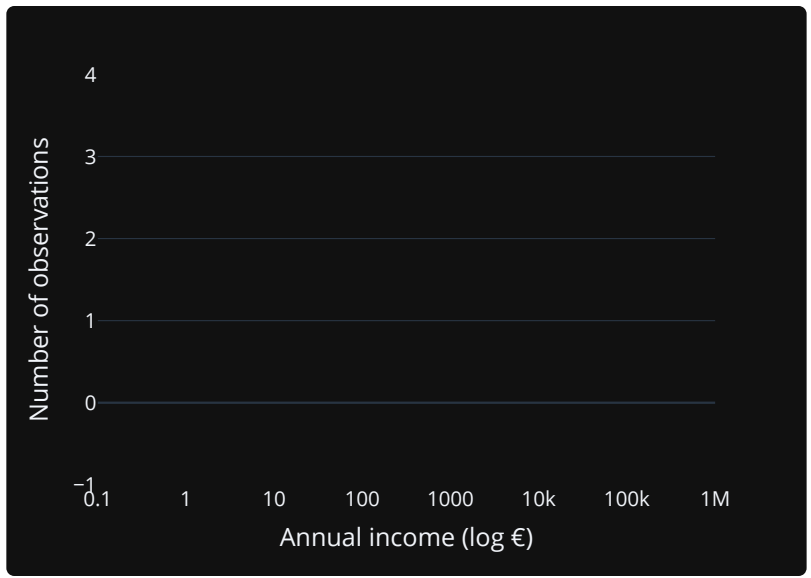
Histogram: Linear

```
fig = income.plot.hist(nbins=100)
fig.update_layout(
    showlegend=False,
    yaxis_title="Number of observations",
    xaxis_title="Annual income (€)",
)
fig.show()
```



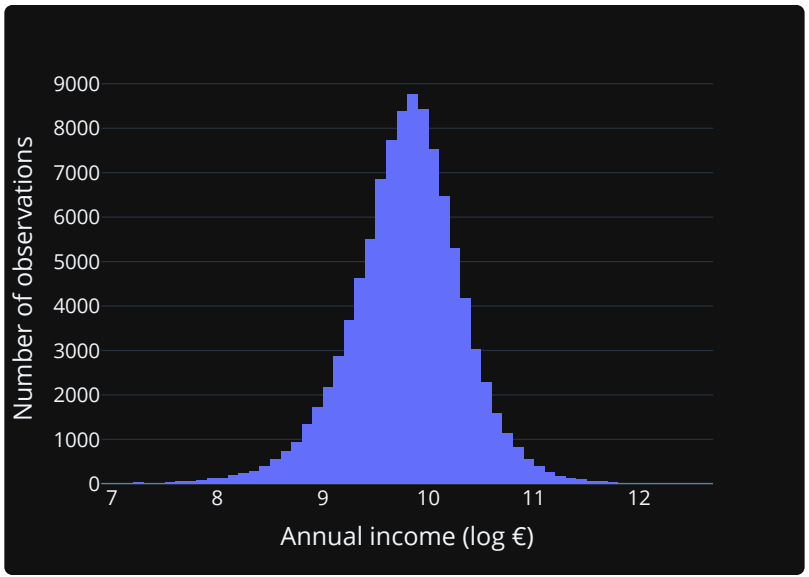
Histogram with automatic log scale

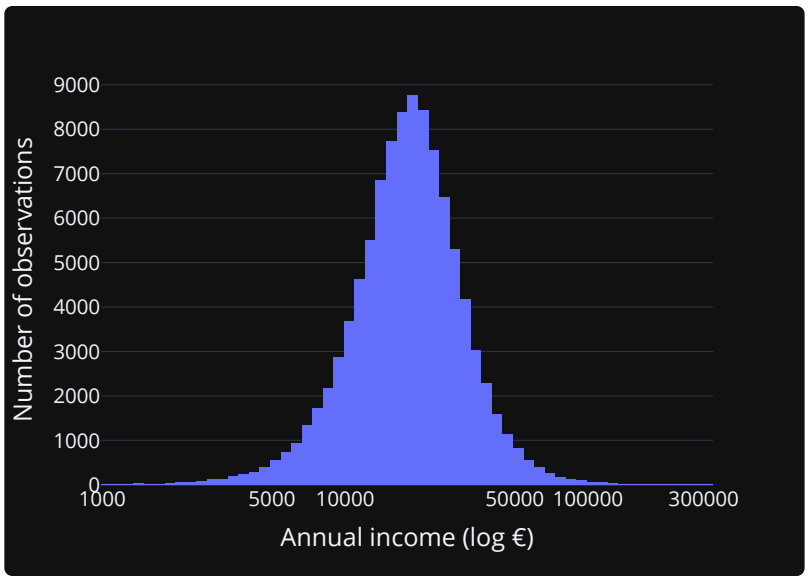
```
fig = income.plot.hist(nbins=100)
fig.update_layout(
    showlegend=False,
    yaxis_title="Number of observations",
    xaxis_title="Annual income (log €)",
    xaxis_type="log",
)
fig.show()
```



Histogram with non-human labels

```
fig = np.log(income).hist(nbins=100)
fig.update_layout(
    showlegend=False,
    yaxis_title="Number of observations",
    xaxis_title="Annual income (log €)",
)
fig.show()
```



Histogram with manual labels

```
xtickvals = pd.Series([1000, 5_000, 10_000
fig = np.log(income).hist(nbins=100)
fig.update_layout(
    showlegend=False,
    yaxis_title="Number of observations",
    xaxis_title="Annual income (log €)",
    xaxis_tickvals=np.log(xtickvals),
    xaxis_ticktext=xtickvals,
)
fig.show()
```