



# Empirical Cumulative Distribution Plots in Python

How to add empirical cumulative distribution function (ECDF) plots.

plots

Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. [Sign up for early access now.](https://plotly.com/studio/?utm_medium=graphing-libraries&utm_campaign=studio_early_access&utm_content=sidebar) ([https://plotly.com/studio/?utm\\_medium=graphing-libraries&utm\\_campaign=studio\\_early\\_access&utm\\_content=sidebar](https://plotly.com/studio/?utm_medium=graphing-libraries&utm_campaign=studio_early_access&utm_content=sidebar))

## Overview

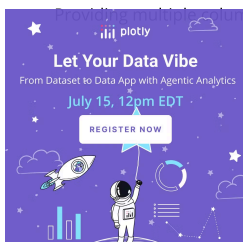
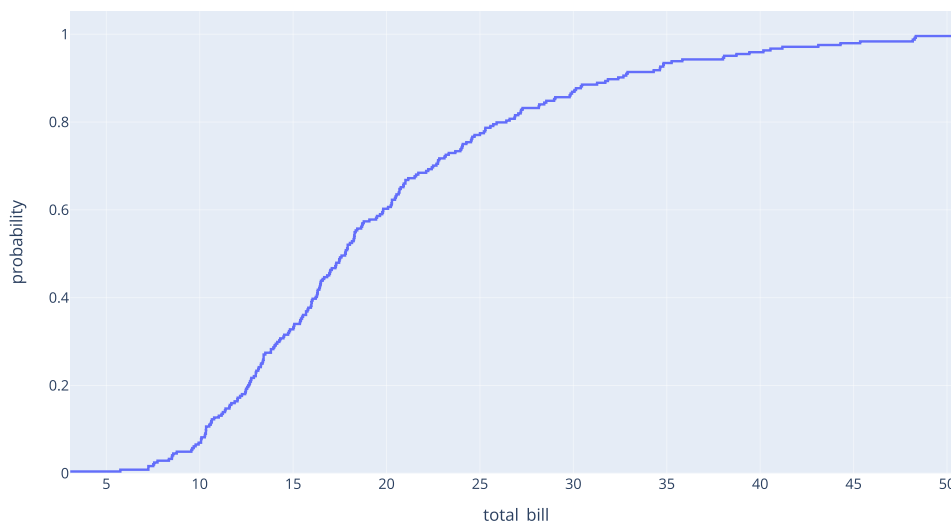
[Empirical cumulative distribution function plots](https://en.wikipedia.org/wiki/Empirical_distribution_function) ([https://en.wikipedia.org/wiki/Empirical\\_distribution\\_function](https://en.wikipedia.org/wiki/Empirical_distribution_function)) are a way to visualize the distribution of a variable, and Plotly Express has a built-in function, `px.ecdf()` to generate such plots. [Plotly Express](#) (/python/plotly-express/) is the easy-to-use, high-level interface to Plotly, which [operates on a variety of types of data](#) (/python/px-arguments/) and produces [easy-to-style figures](#) (/python/styling-plotly-express/).

Alternatives to ECDF plots for visualizing distributions include [histograms](https://plotly.com/python/histograms/) (<https://plotly.com/python/histograms/>), [violin plots](https://plotly.com/python/violin/) (<https://plotly.com/python/violin/>), [box plots](https://plotly.com/python/box-plots/) (<https://plotly.com/python/box-plots/>) and [strip charts](https://plotly.com/python/strip-charts/) (<https://plotly.com/python/strip-charts/>).

## Simple ECDF Plots

Providing a single column to the x variable yields a basic ECDF plot.

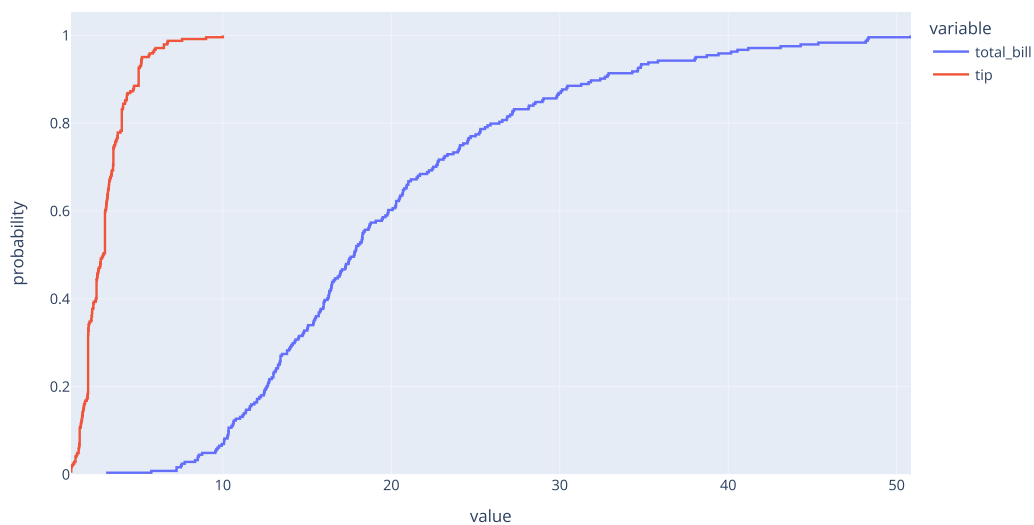
```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill")
fig.show()
```



Plotly's [Plotly Express](#) [wide-form data support](#) (<https://plotly.com/python/wide-form/>) to show multiple variables on the same plot.

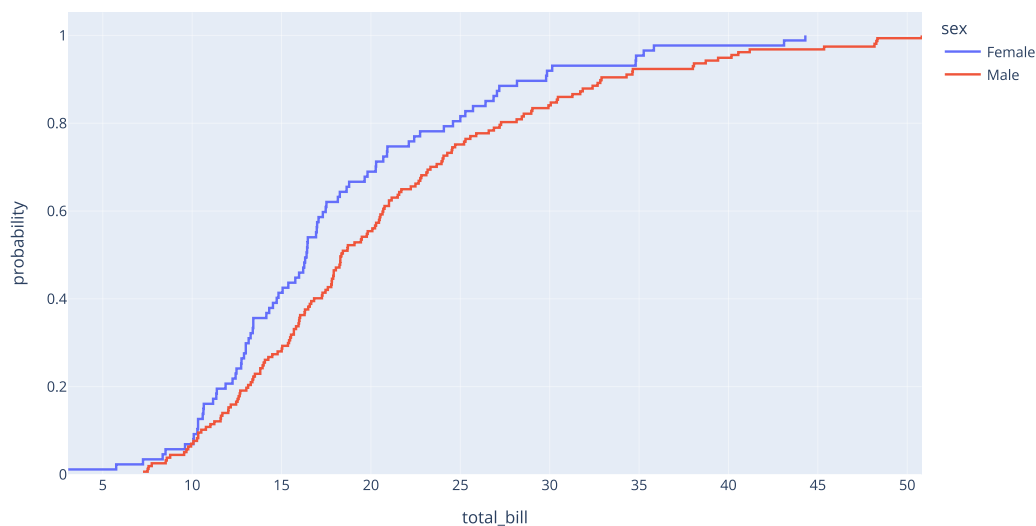
```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x=["total_bill", "tip"])
fig.show()
```

plots

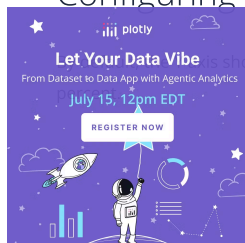


It is also possible to map another variable to the color dimension of a plot.

```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex")
fig.show()
```



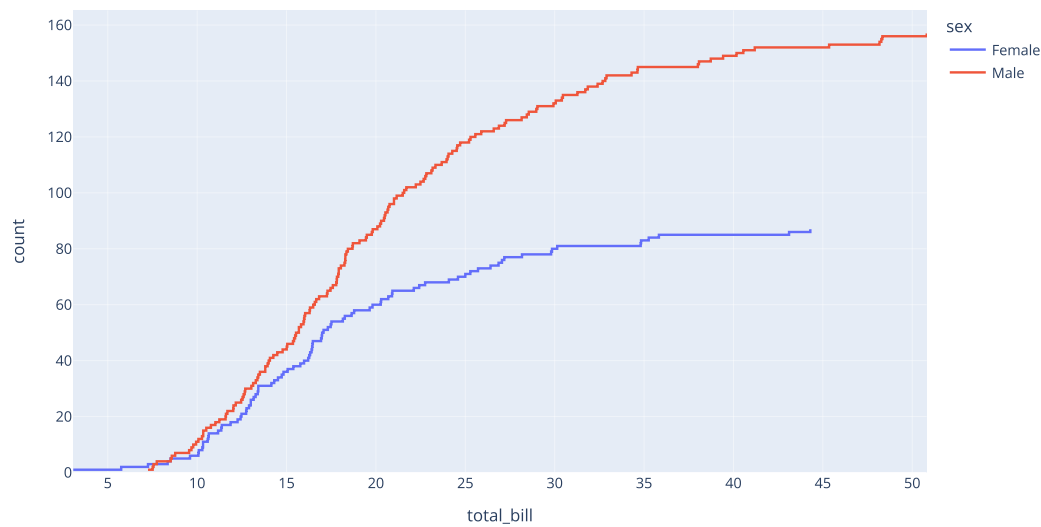
## Configuring the Y axis



shows probability, but it is also possible to show raw counts by setting the `ecdfnorm` argument to `None` or to show percentages by setting it to

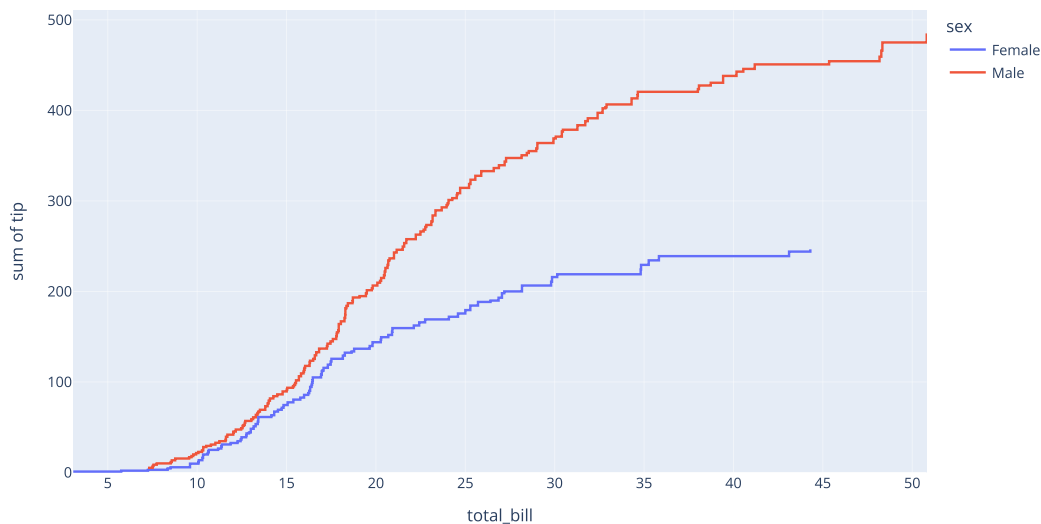
```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", ecdfnorm=None)
fig.show()
```

plots

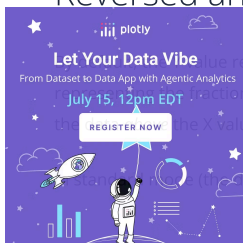


If a y value is provided, the Y axis is set to the sum of y rather than counts.

```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", y="tip", color="sex", ecdfnorm=None)
fig.show()
```



## Reversed and Complementary CDF plots



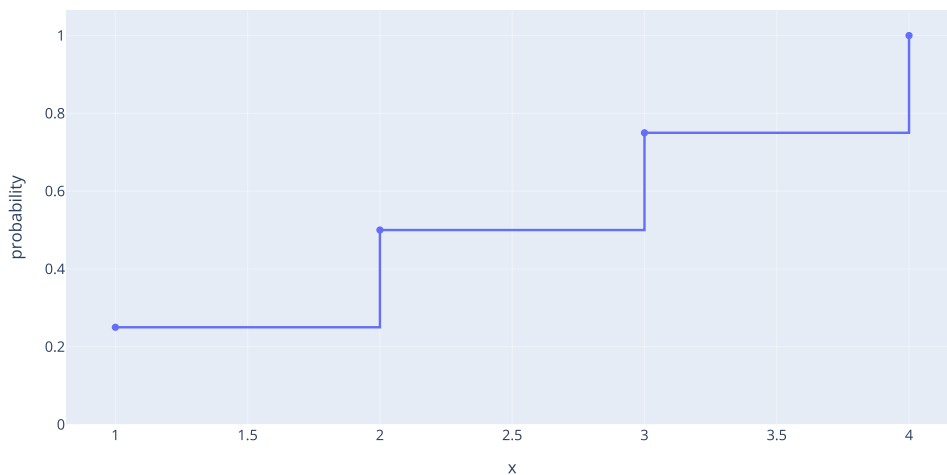
represents the fraction of the data that is *at or below* the value on the X axis. Setting `ecdfmode` to "reversed" reverses this, with the Y axis representing the fraction of the data *at or above* the X value. Setting `ecdfmode` to "complementary" plots 1-ECDF, meaning that the Y values represent the fraction of the data that is *not* at or below the X value.

By default, the right-most point is at 1 (or the total count/sum, depending on `ecdfnorm`) and the right-most point is above 0.

```
import plotly.express as px
fig = px.ecdf(df, x=[1,2,3,4], markers=True, ecdfmode="standard",
              title="ecdfmode='standard' (Y=fraction at or below X value, this the default)")
fig.show()
```

ecdfmode='standard' (Y=fraction at or below X value, this the default)

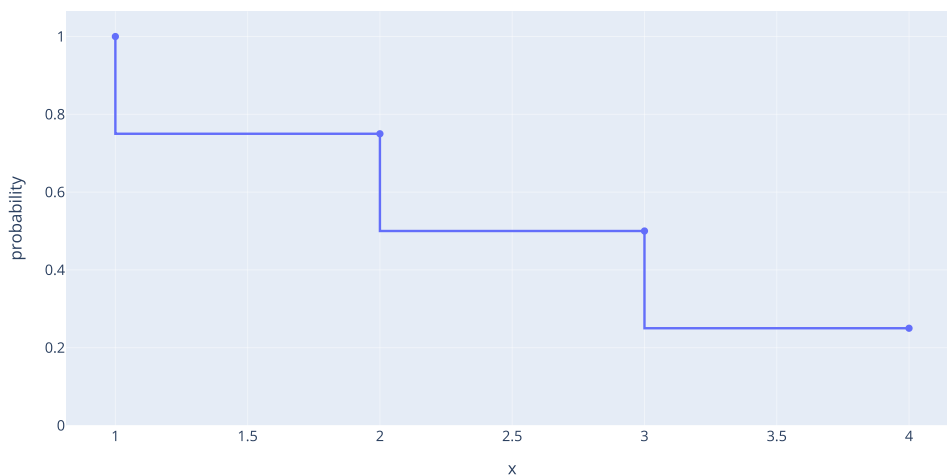
plots



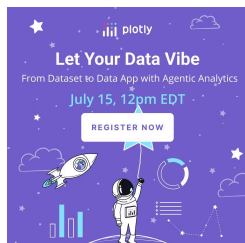
In reversed mode, the right-most point is at 1 (or the total count/sum, depending on ecdfnorm) and the left-most point is above 0.

```
import plotly.express as px
fig = px.ecdf(df, x=[1,2,3,4], markers=True, ecdfmode="reversed",
              title="ecdfmode='reversed' (Y=fraction at or above X value)")
fig.show()
```

ecdfmode='reversed' (Y=fraction at or above X value)



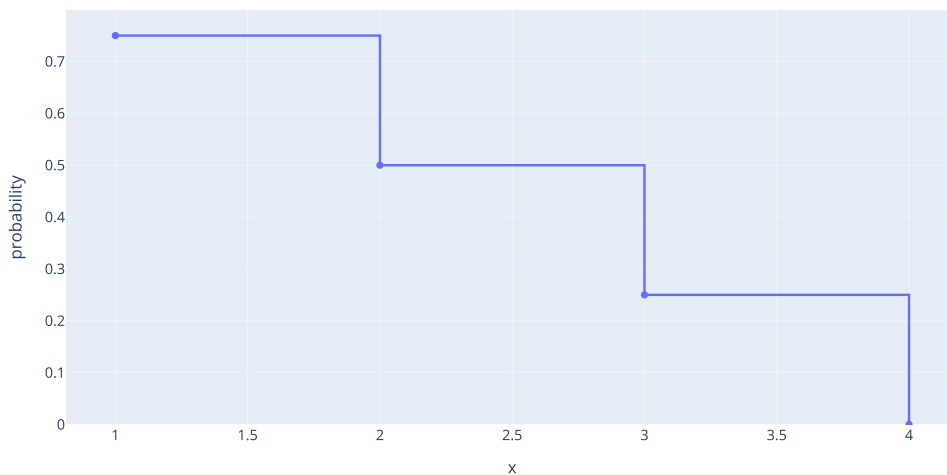
In complementary mode, the right-most point is at 0 and no points are at 1 (or the total count/sum) per the definition of the CCDF as 1-ECDF, which has no point at 0.



```
import plotly.express as px
fig = px.ecdf(df, x=[1,2,3,4], markers=True, ecdfmode="complementary",
              title="ecdfmode='complementary' (Y=fraction above X value)")
fig.show()
```

ecdfmode='complementary' (Y=fraction above X value)

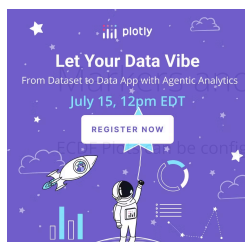
plots



## Orientation

By default, plots are oriented vertically (i.e. the variable is on the X axis and counted/summed upwards), but this can be overridden with the orientation argument.

```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", y="tip", color="sex", ecdfnorm=None, orientation="h")
fig.show()
```

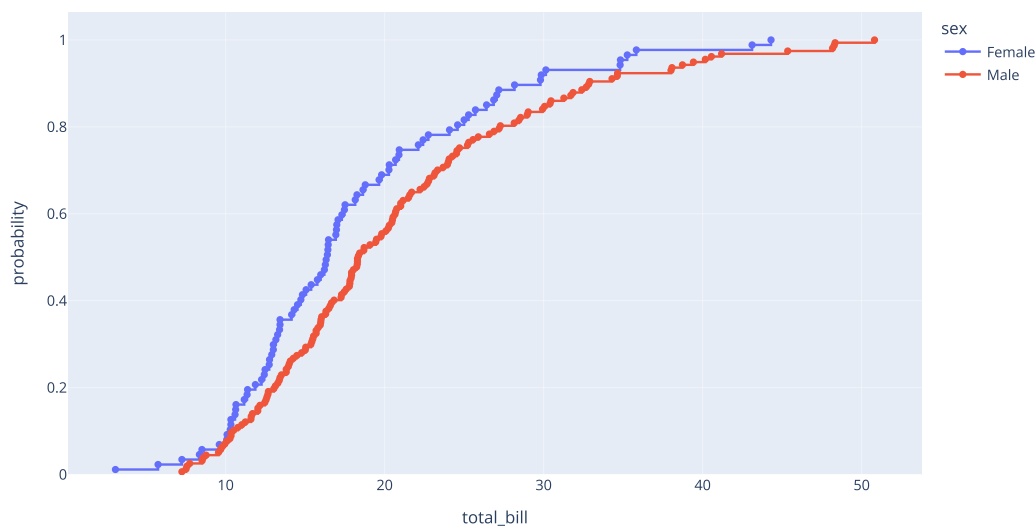


/or Lines

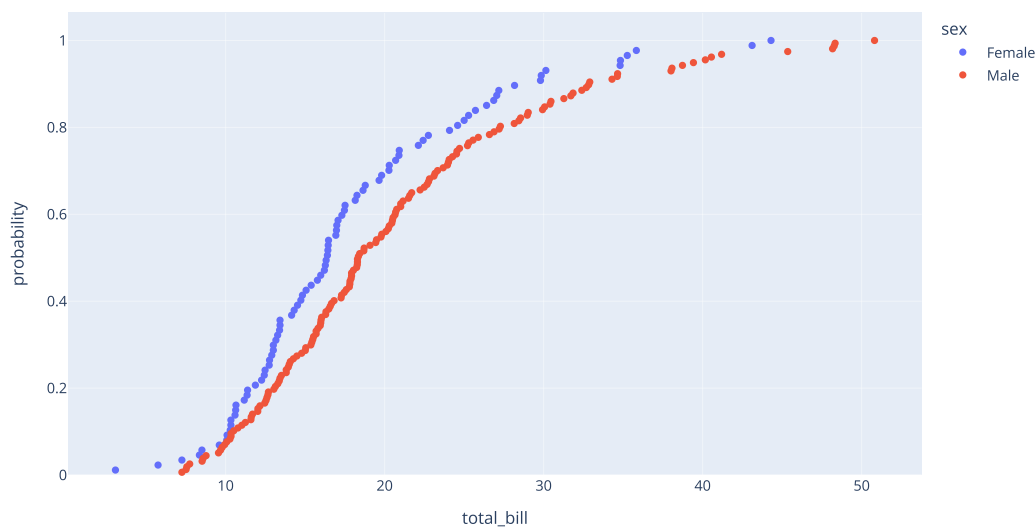
be configured to show lines and/or markers.

```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", markers=True)
fig.show()
```

plots

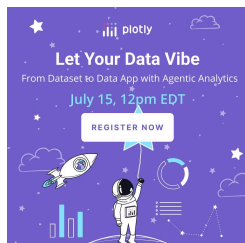


```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", markers=True, lines=False)
fig.show()
```



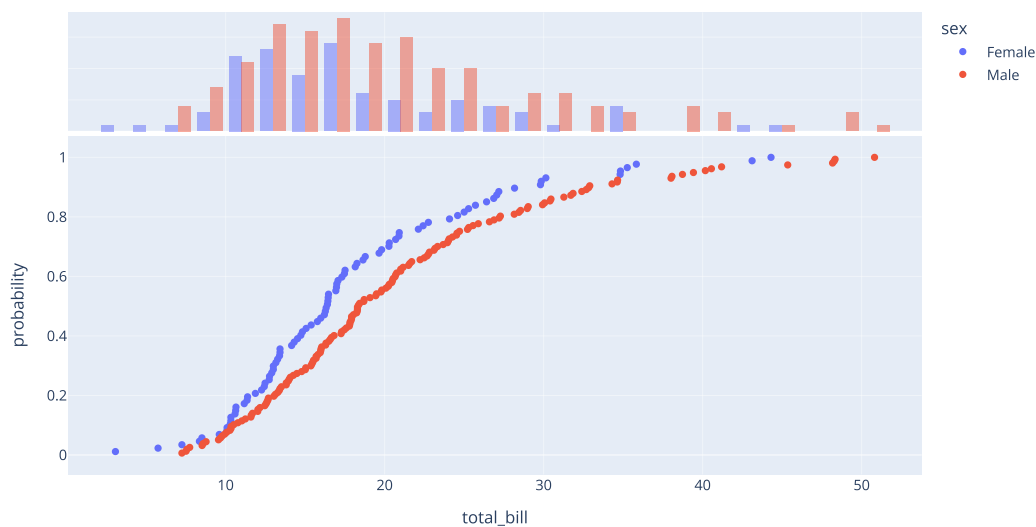
## Marginal Plots

ECDF plots also support [marginal plots](https://plotly.com/python/marginal-plots/) (<https://plotly.com/python/marginal-plots/>)

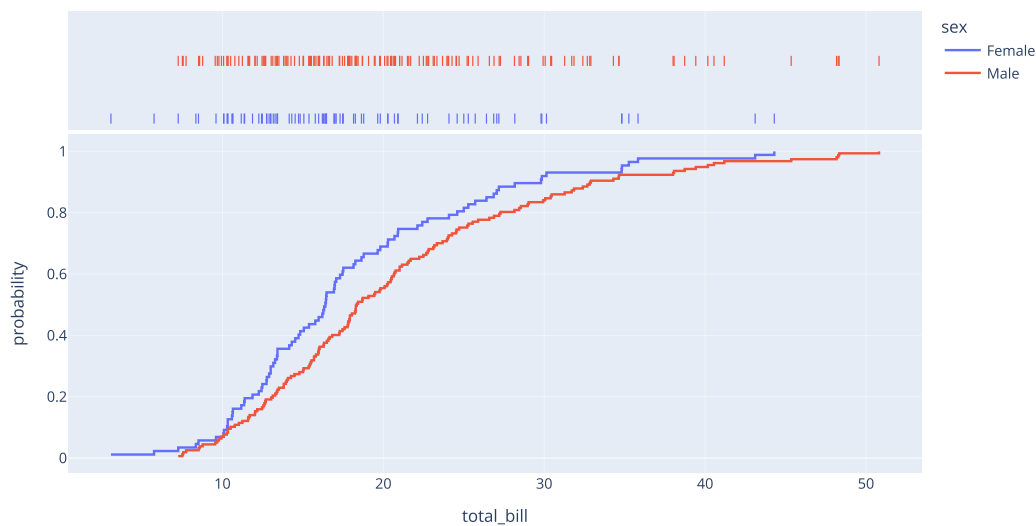


```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", markers=True, lines=False, marginal="histogram")
fig.show()
```

plots

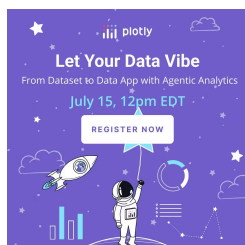


```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", marginal="rug")
fig.show()
```



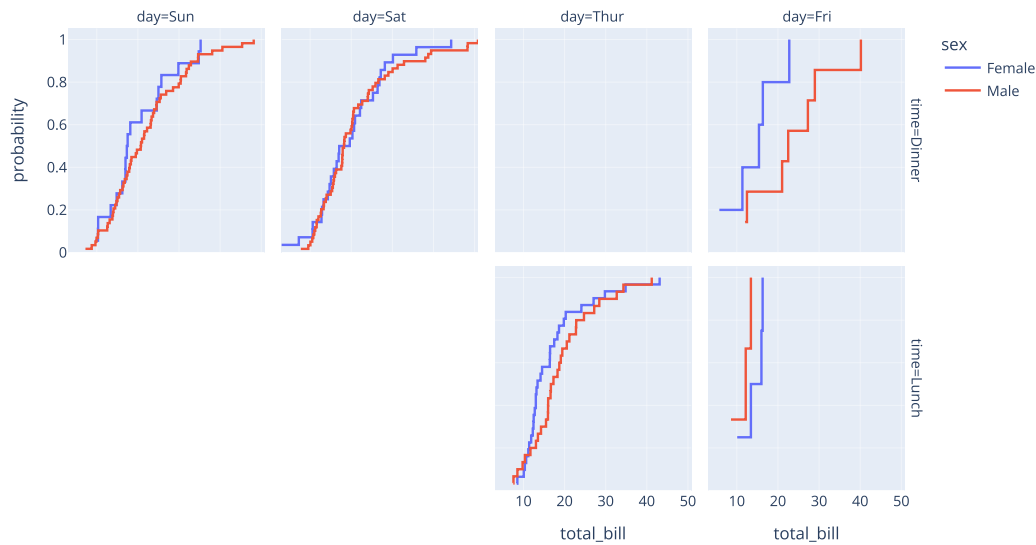
## Facets

ECDF Plots also support [faceting](https://plotly.com/python/facet-plots/) (<https://plotly.com/python/facet-plots/>)



```
import plotly.express as px
df = px.data.tips()
fig = px.ecdf(df, x="total_bill", color="sex", facet_row="time", facet_col="day")
fig.show()
```

plots



## What About Dash?

[Dash](https://dash.plot.ly/) (<https://dash.plot.ly/>) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at <https://dash.plot.ly/installation> (<https://dash.plot.ly/installation>).

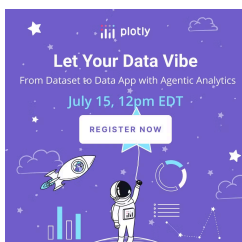
Everywhere in this page that you see `fig.show()`, you can display the same figure in a Dash application by passing it to the figure argument of the [Graph component](https://dash.plot.ly/dash-core-components/graph) (<https://dash.plot.ly/dash-core-components/graph>) from the built-in `dash_core_components` package like this:

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )


from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False) # Turn off reloader if inside Jupyter
```







# Dash your way to interactive web apps.

No JavaScript required!

GET STARTED NOW

### My First App with Data, Graph, and Controls


pop

lifeExp

gdpPerCap

country	pop	continent	lifeExp	gdpPerCap
Afghanistan	31889923	Asia	43.828	974.5883384
Albania	2600522	Europe	76.422	5937.029525999999
Algeria	33333216	Africa	72.361	6223.367465
Angola	12420676	Africa	42.731	4707.231267
Argentina	40301927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36126.4927
Bahrain	708573	Asia	75.635	29796.04854
Bangladesh	150448359	Asia	64.062	1501.253792
Belgium	10592226	Europe	79.441	33692.04908
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137884

1 / 12



([https://dash.plotly.com/tutorial?utm\\_medium=graphing\\_libraries&utm\\_content=python\\_footer](https://dash.plotly.com/tutorial?utm_medium=graphing_libraries&utm_content=python_footer))

plots

JOIN OUR MAILING LIST

Sign up to stay in the loop with all things Plotly — from Dash Club to product updates, webinars, and more!

SUBSCRIBE  
(<https://go.plot.ly/subscription>)

About Us

Careers (<https://plotly.com/careers>)  
Resources (<https://plotly.com/resources/>)  
Blog (<https://medium.com/@plotlygraphs>)

Products

Dash (<https://plotly.com/dash/>)  
Consulting and Training  
(<https://plotly.com/consulting-and-oem/>)

Support

Community Support (<https://community.plot.ly/>)  
Documentation (<https://plotly.com/graphing-libraries>)

Pricing

Enterprise Pricing (<https://plotly.com/get-pricing/>)

Copyright © 2025 Plotly. All rights reserved.

[Terms of Service \(https://community.plotly.com/tos\)](https://community.plotly.com/tos) [Privacy Policy \(https://plotly.com/privacy/\)](https://plotly.com/privacy/)

