Funnel Chart in Python

How to make funnel-chart plots in Python with Plotly.

Introduction

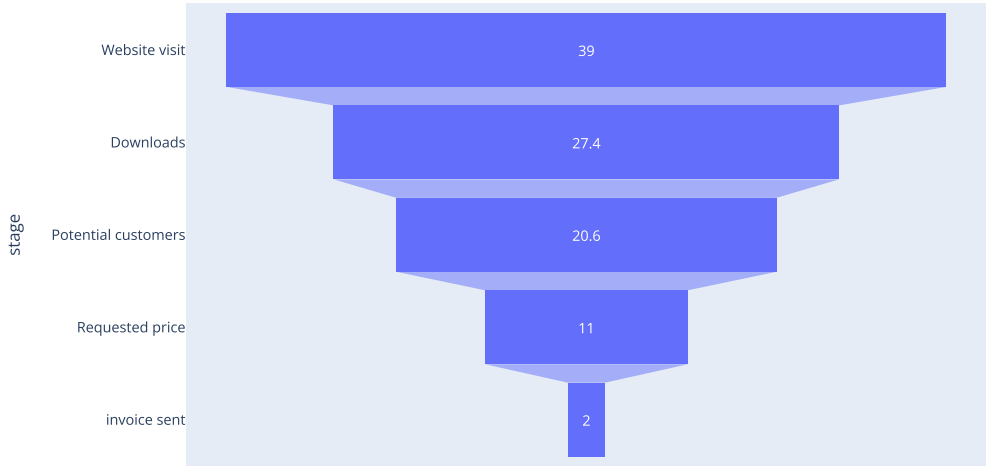
Funnel charts are often used to represent data in different stages of a business process. It's an important mechanism in Business Intelligence to identify potential problem areas of a process. For example, it's used to observe the revenue or loss in a sales process for each stage, and displays values that are decreasing progressively. Each stage is illustrated as a percentage of the total of all values.

Basic Funnel Plot with plotly.express

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/).

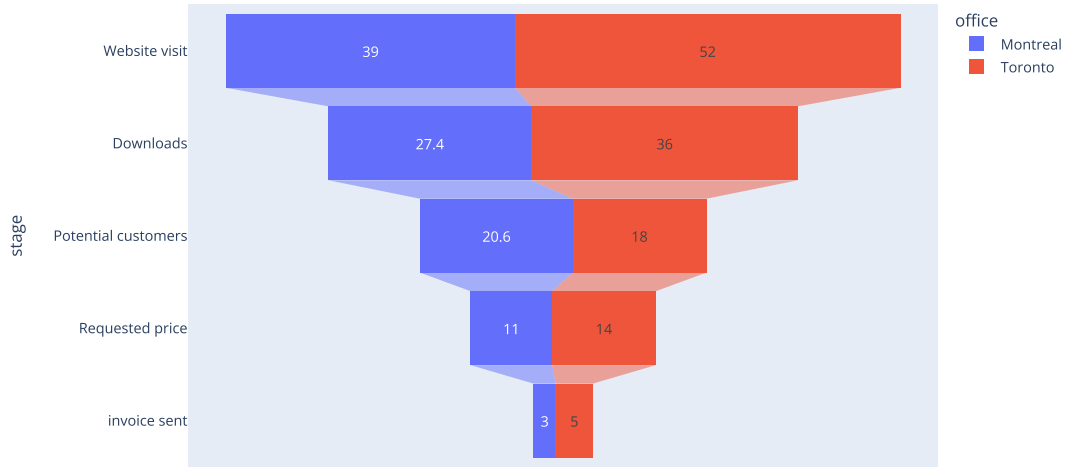
With px.funnel, each row of the DataFrame is represented as a stage of the funnel.

```
import plotly.express as px
data = dict(
    number=[39, 27.4, 20.6, 11, 2],
    stage=["Website visit", "Downloads", "Potential customers", "Requested price", "invoice sent"])
fig = px.funnel(data, x='number', y='stage')
fig.show()
```



Stacked Funnel Plot with plotly.express

```
import plotly.express as px
import pandas as pd
stages = ["Website visit", "Downloads", "Potential customers", "Requested price", "invoice sent"]
df_mtl = pd.DataFrame(dict(number=[39, 27.4, 20.6, 11, 3], stage=stages))
df_mtl['office'] = 'Montreal'
df_toronto = pd.DataFrame(dict(number=[52, 36, 18, 14, 5], stage=stages))
df_toronto['office'] = 'Toronto'
df = pd.concat([df_mtl, df_toronto], axis=0)
fig = px.funnel(df, x='number', y='stage', color='office')
fig.show()
```



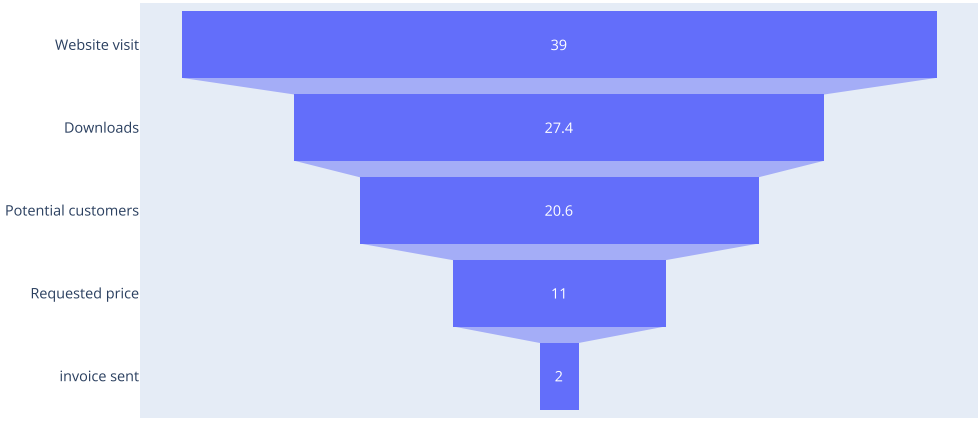
Basic Funnel Chart with graph_objects trace go.Funnel

If Plotly Express does not provide a good starting point, it is also possible to use the more generic go.Funnel class from plotly.graph_objects (/python/graph-objects/).

```
from plotly import graph_objects as go

fig = go.Figure(go.Funnel(
    y = ["Website visit", "Downloads", "Potential customers", "Requested price", "invoice sent"],
    x = [39, 27.4, 20.6, 11, 2]))

fig.show()
```

s
ess
ts trace

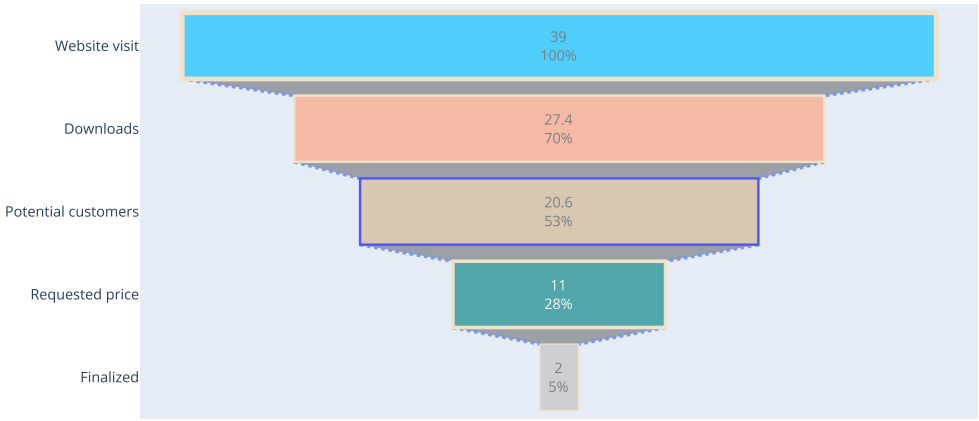
cpress
elarea
unnel



Setting Marker Size and Color

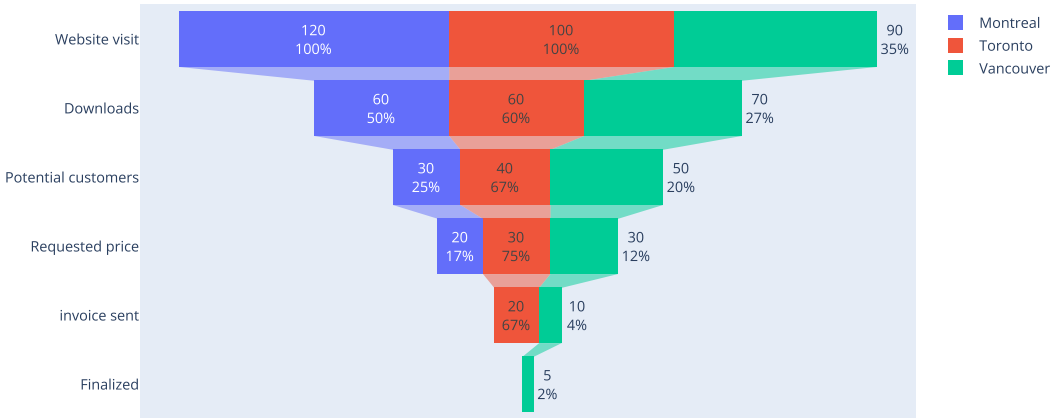
This example uses textposition (https://plotly.com/python/reference/scatter/#scatter-textposition) and textinfo (https://plotly.com/python/reference/funnel/#funnel-textinfo) to determine information appears on the graph, and shows how to customize the bars.

```
from plotly import graph_objects as go

fig = go.Figure(go.Funnel(
    y = ["Website visit", "Downloads", "Potential customers", "Requested price", "Finalized"],
    x = [39, 27.4, 20.6, 11, 2],
    textposition = "inside",
    textinfo = "value+percent initial",
    opacity = 0.65, marker = {"color": ["deepskyblue", "lightsalmon", "tan", "teal", "silver"],
    "line": {"width": [4, 2, 2, 3, 1, 1], "color": ["wheat", "wheat", "blue", "wheat", "wheat"]}},
    connector = {"line": {"color": "royalblue", "dash": "dot", "width": 3}})
    )

fig.show()
```



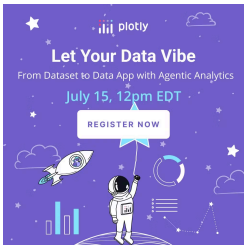
Stacked Funnel Plot with go.Funnel

```
from plotly import graph_objects as go

fig = go.Figure()

fig.add_trace(go.Funnel(
    name = 'Montreal',
    y = ["Website visit", "Downloads", "Potential customers", "Requested price"],
    x = [120, 60, 30, 20],
    textinfo = "value+percent initial"))

fig.add_trace(go.Funnel(
    name = 'Toronto',
    orientation = "h",
    y = ["Website visit", "Downloads", "Potential customers", "Requested price", "invoice sent"],
    x = [100, 60, 40, 30, 20],
    textposition = "inside",
    textinfo = "value+percent previous"))

fig.add_trace(go.Funnel(
    name = 'Vancouver',
    orientation = "h",
    y = ["Website visit", "Downloads", "Potential customers", "Requested price", "invoice sent", "Finalized"],
    x = [90, 70, 50, 30, 10, 5],
    textposition = "outside",
    textinfo = "value+percent total"))

fig.show()
```

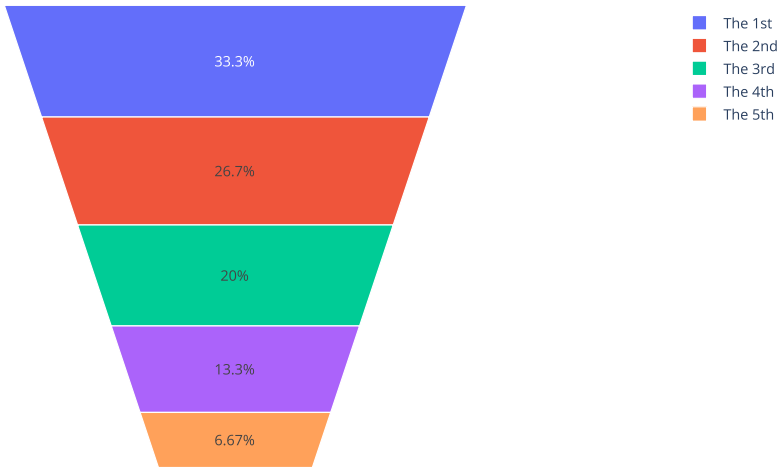


Basic Area Funnel Plot with plotly.express

With px.funnel_area, each row of the DataFrame is represented as a stage of the funnel.



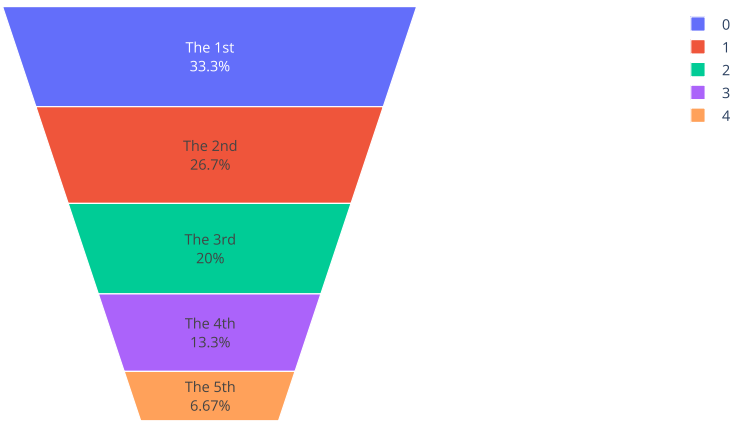
```
import plotly.express as px
fig = px.funnel_area(names=["The 1st","The 2nd", "The 3rd", "The 4th", "The 5th"],
                     values=[5, 4, 3, 2, 1])
fig.show()
```



Basic Area Funnel Plot with go.Funnelarea

If Plotly Express does not provide a good starting point, it is also possible to use the more generic go.Funnelarea class from plotly.graph_objects (/python/graph-objects/).

```
from plotly import graph_objects as go

fig = go.Figure(go.Funnelarea(
    text = ["The 1st","The 2nd", "The 3rd", "The 4th", "The 5th"],
    values = [5, 4, 3, 2, 1]
    ))
fig.show()
```

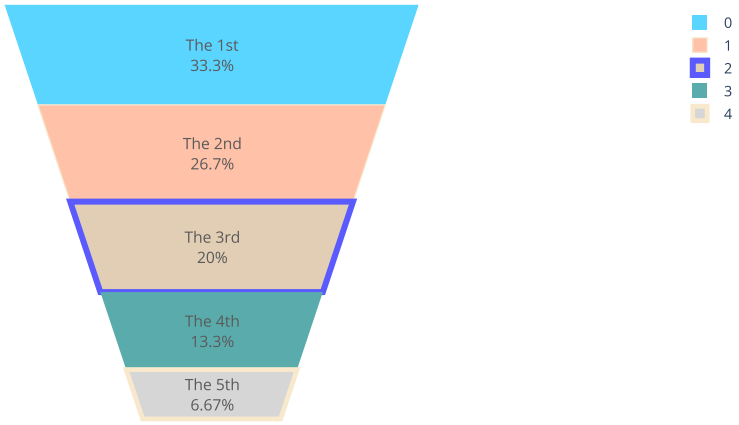


and Color in Area Funnel Plots

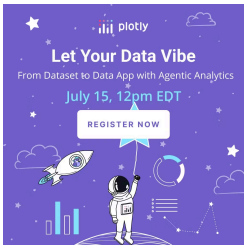
```
from plotly import graph_objects as go

fig = go.Figure(go.Funnelarea(
      values = [5, 4, 3, 2, 1], text = ["The 1st","The 2nd", "The 3rd", "The 4th", "The 5th"],
      marker = {"colors": ["deepskyblue", "lightsalmon", "tan", "teal", "silver"],
                "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat"], "width": [0, 1, 5, 0, 4]}},
      textfont = {"family": "Old Standard TT, serif", "size": 13, "color": "black"}, opacity = 0.65))
fig.show()
```

s
ess
ts trace

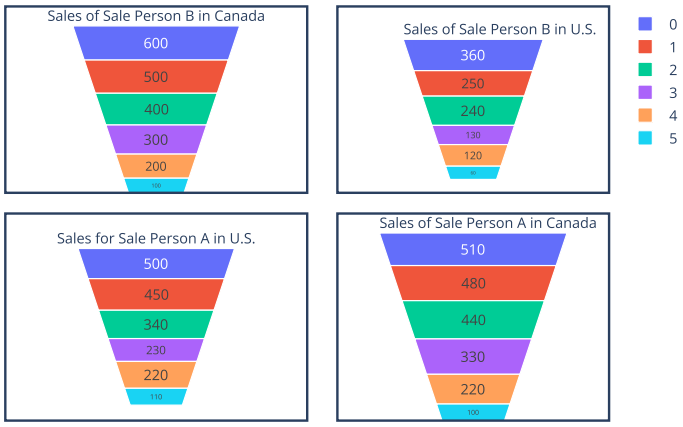
press
elarea
unnel



Multiple Area Funnels



```
from plotly import graph_objects as go

fig = go.Figure()

fig.add_trace(go.Funnelarea(
    scalegroup = "first", values = [500, 450, 340, 230, 220, 110], textinfo = "value",
    title = {"position": "top center", "text": "Sales for Sale Person A in U.S."},
    domain = {"x": [0, 0.5], "y": [0, 0.5]}))

fig.add_trace(go.Funnelarea(
    scalegroup = "first", values = [600, 500, 400, 300, 200, 100], textinfo = "value",
    title = {"position": "top center", "text": "Sales of Sale Person B in Canada"},
    domain = {"x": [0, 0.5], "y": [0.55, 1]}))

fig.add_trace(go.Funnelarea(
    scalegroup = "second", values = [510, 480, 440, 330, 220, 100], textinfo = "value",
    title = {"position": "top left", "text": "Sales of Sale Person A in Canada"},
    domain = {"x": [0.55, 1], "y": [0, 0.5]}))

fig.add_trace(go.Funnelarea(
            scalegroup = "second", values = [360, 250, 240, 130, 120, 60],
            textinfo = "value", title = {"position": "top left", "text": "Sales of Sale Person B in U.S."},
            domain = {"x": [0.55, 1], "y": [0.55, 1]}))

fig.update_layout(
            margin = {"l": 200, "r": 200}, shapes = [
            {"x0": 0, "x1": 0.5, "y0": 0, "y1": 0.5},
            {"x0": 0, "x1": 0.5, "y0": 0.55, "y1": 1},
            {"x0": 0.55, "x1": 1, "y0": 0, "y1": 0.5},
            {"x0": 0.55, "x1": 1, "y0": 0.55, "y1": 1}])

fig.show()
```

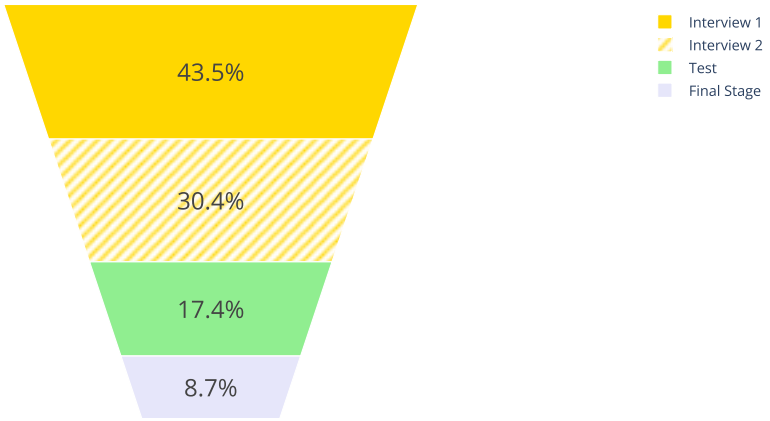


Pattern Fills

New in 5.15

Funnel area charts support patterns (/python/pattern-hatching-texture/) (also known as hatching or texture) in addition to color. In this example, we add a pattern to the second stage of the funnel.

```
from plotly import graph_objects as go

colors = ["gold", "gold", "lightgreen", "lavender"]

fig = go.Figure(
    go.Funnelarea(
        labels=["Interview 1", "Interview 2", "Test", "Final Stage"],
        values=[100, 70, 40, 20],
        textfont_size=20,
        marker=dict(colors=colors, pattern=dict(shape=["", "/", "", ""])),
    )
)
fig.show()
```



Reference

See function reference for px.(funnel) (https://plotly.com/python-api-reference/generated/plotly.express.funnel) or https://plotly.com/python/reference/funnel/ (https://plotly.com/python/reference/funnel/) and https://plotly.com/python/reference/funnelarea/ (https://plotly.com/python/reference/funnelarea/) for more information and chart attribute options!

What About Dash?

Dash (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) 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

                                    v([
                                    =fig)


                                    use_reloader=False)  # Turn off reloader if inside Jupyter
```

s
ess
ts trace