typst-ribbons Examples

A gallery of practical examples

2025-11-07

Contents

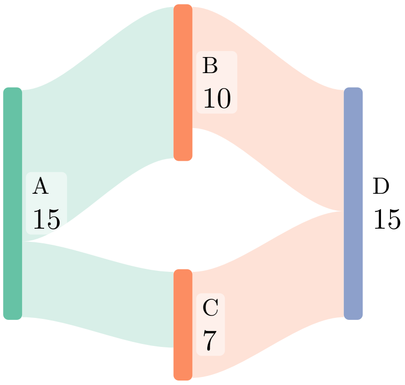
1 Basic Examples . . . 3
1.1 Simple Sankey Diagram . . . 3
1.2 Simple Chord Diagram . . . 4
2 Layout Examples . . . 5
2.1 Vertical Sankey Diagram . . . 5
2.2 Directed Chord Diagram . . . 6
2.3 Compact Sankey . . . 7
3 Color and Style Examples . . . 8
3.1 Layer-Based Coloring . . . 8
3.2 Custom Node Colors . . . 9
3.3 Gradient Ribbons . . . 10
3.4 Categorical Coloring . . . 11
4 Practical Use Cases . . . 12
4.1 Budget Breakdown . . . 12
4.2 Energy Flow . . . 13
4.3 User Journey . . . 14
4.4 Trade Flows . . . 15
5 Advanced Styling . . . 16
5.1 Custom Labels . . . 16
5.2 Sharp Corners . . . 17
5.3 Multiple Edges . . . 18
6 Data Format Examples . . . 19
6.1 Adjacency Dictionary . . . 19
6.2 Adjacency List . . . 19
6.3 Adjacency Matrix . . . 19
7 Color Palette Showcase . . . 20
7.1 ColorBrewer Palette . . . 20
7.2 Tableau Palette . . . 21
7.3 Catppuccin Palette . . . 22

1 Basic Examples

1.1 Simple Sankey Diagram

A minimal example showing basic flow from sources to targets.

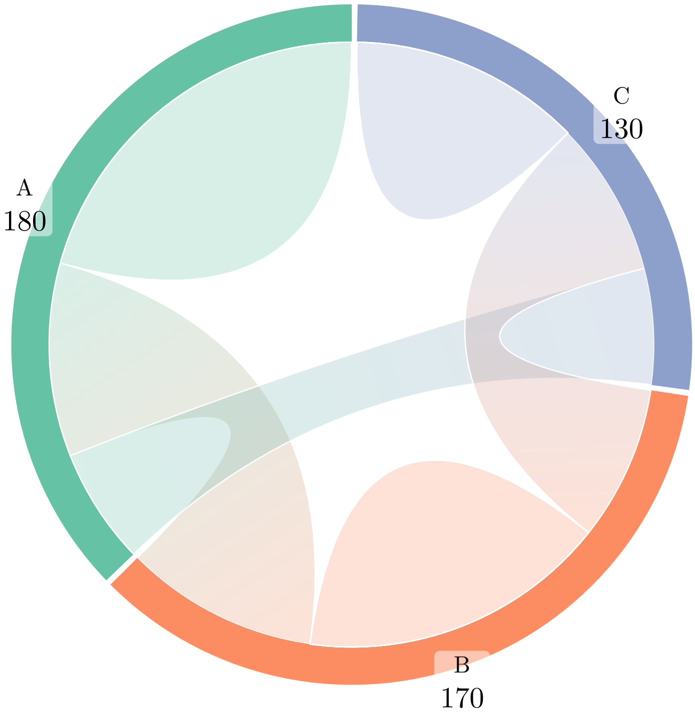
```
#sankey-diagram((
  "A": ("B": 10, "C": 5),
  "B": ("D": 8),
  "C": ("D": 7),
))
```



1.2 Simple Chord Diagram

A basic circular diagram showing relationships.

```
#chord-diagram((
  "A": ("A": 100, "B": 50, "C": 30),
  "B": ("A": 50, "B": 80, "C": 40),
  "C": ("A": 30, "B": 40, "C": 60),
))
```

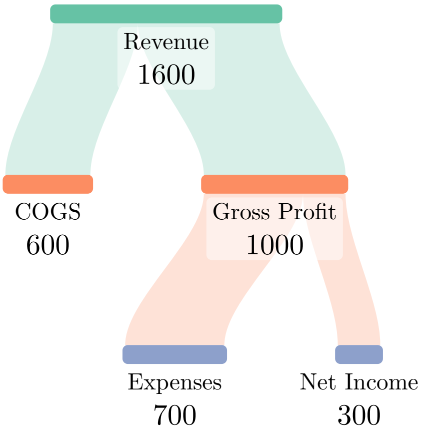


2 Layout Examples

2.1 Vertical Sankey Diagram

Top-to-bottom flow instead of left-to-right.

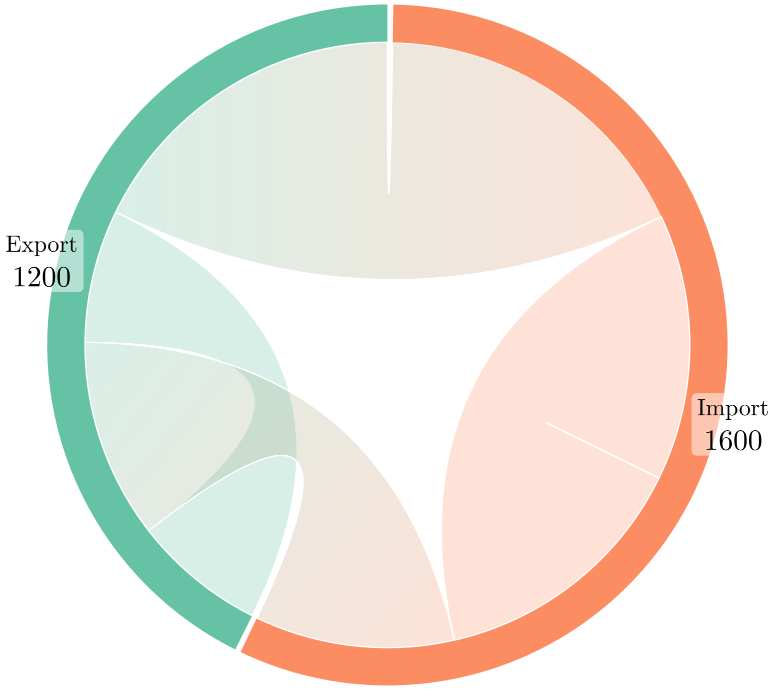
```
#sankey-diagram(
  (
    "Revenue": ("Gross Profit": 1000, "COGS": 600),
    "Gross Profit": ("Net Income": 300, "Expenses": 700),
  ),
  layout: layout.auto-linear(vertical: true)
)
```



2.2 Directed Chord Diagram

Showing asymmetric flow in circular layout.

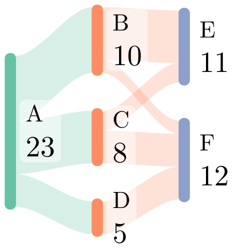
```
#chord-diagram(
  (
    "Export": ("Import": 500, "Export": 200),
    "Import": ("Export": 300, "Import": 400),
  ),
  layout: layout.circular(directed: true)
)
```



2.3 Compact Sankey

Tight spacing for fitting more in less space.

```
#sankey-diagram(
  (
    "A": ("B": 10, "C": 8, "D": 5),
    "B": ("E": 8, "F": 2),
    "C": ("E": 3, "F": 5),
    "D": ("F": 5),
  ),
  layout: layout.auto-linear(
    layer-gap: 1,
    node-gap: 0.5,
    node-width: 0.15,
    base-node-height: 2,
  )
)
```

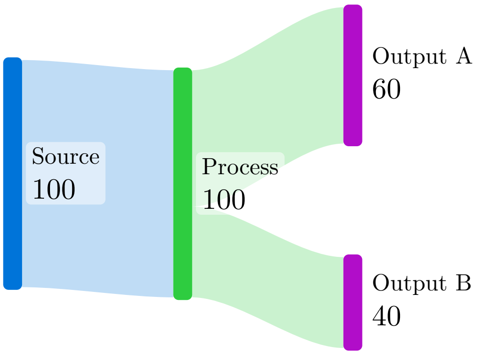


3 Color and Style Examples

3.1 Layer-Based Coloring

Different color for each layer.

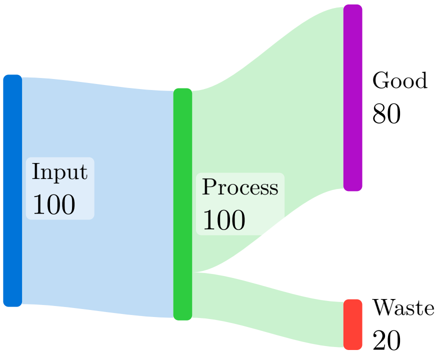
```
#sankey-diagram(
  (
    "Source": ("Process": 100),
    "Process": ("Output A": 60, "Output B": 40),
  ),
  tinter: tinter.layer-tinter(
    palette: (blue, green, purple)
  )
)
```



3.2 Custom Node Colors

Manually specified colors for each node.

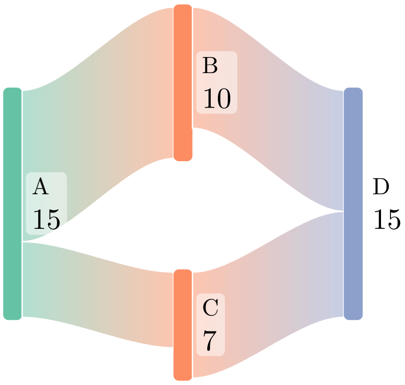
```
#sankey-diagram(
  (
    "Input": ("Process": 100),
    "Process": ("Good": 80, "Waste": 20),
  ),
  tinter: tinter.dict-tinter((
    "Input": blue,
    "Process": green,
    "Good": purple,
    "Waste": red,
  ))
)
```



3.3 Gradient Ribbons

Ribbons with gradients from source to target.

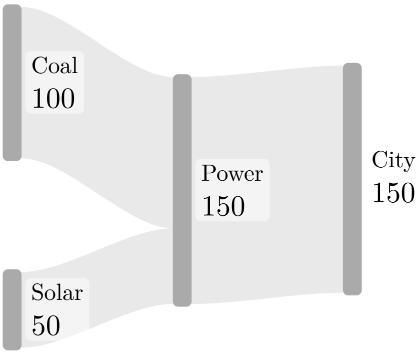
```
#sankey-diagram(
  (
    "A": ("B": 10, "C": 5),
    "B": ("D": 8),
    "C": ("D": 7),
  ),
  ribbon-stylizer: ribbon-stylizer.gradient-from-to(
    transparency: 50%,
    stroke-width: 0.5pt,
    stroke-color: white,
  )
)
```



3.4 Categorical Coloring

Grouping nodes by category.

```
#sankey-diagram(
  (
    "Coal": ("Power": 100),
    "Solar": ("Power": 50),
    "Power": ("City": 150),
  ),
  categories: (
    "Coal": "fossil",
    "Solar": "renewable",
    "Power": "processing",
    "City": "consumption",
  ),
  tinter: tinter.categorical-tinter(
    palette: (gray, green, yellow, purple)
  )
)
```

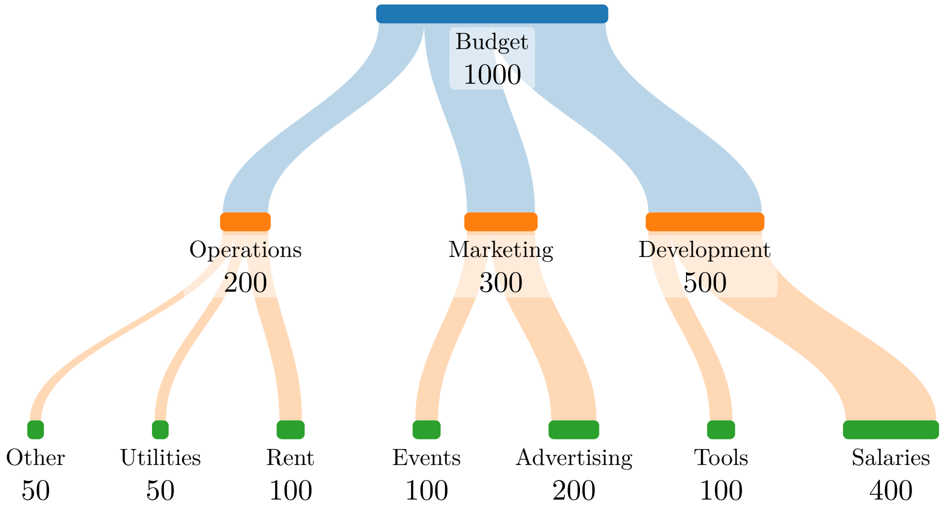


4 Practical Use Cases

4.1 Budget Breakdown

Company budget allocation visualization.

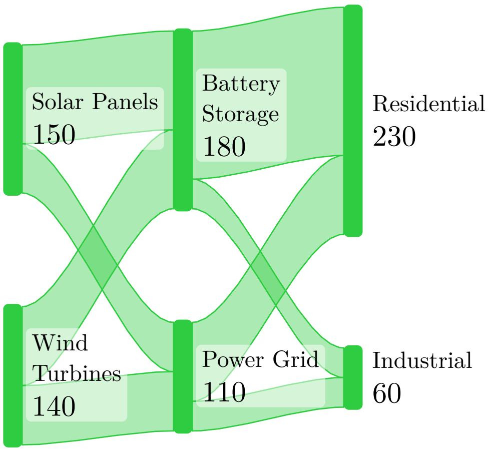
```
#sankey-diagram(
  (
    "Budget": ("Development": 500, "Marketing": 300, "Operations": 200),
    "Development": ("Salaries": 400, "Tools": 100),
    "Marketing": ("Advertising": 200, "Events": 100),
    "Operations": ("Rent": 100, "Utilities": 50, "Other": 50),
  ),
  layout: layout.auto-linear(
    vertical: true,
    layer-gap: 2.5,
    curve-factor: 0.4,
  ),
  tinter: tinter.layer-tinter(
    palette: palette.tableau
  ),
  ribbon-stylizer: ribbon-stylizer.match-from(
    transparency: 70%,
  )
)
```



4.2 Energy Flow

Renewable energy distribution.

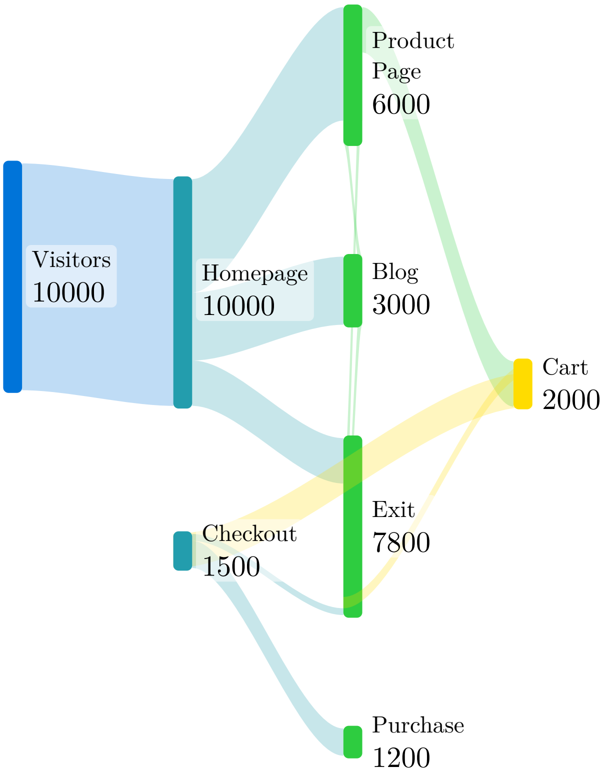
```
#sankey-diagram(
  (
    "Solar": ("Battery": 100, "Grid": 50),
    "Wind": ("Battery": 80, "Grid": 60),
    "Battery": ("Homes": 150, "Industry": 30),
    "Grid": ("Homes": 80, "Industry": 30),
  ),
  aliases: (
    "Solar": "Solar Panels",
    "Wind": "Wind Turbines",
    "Battery": "Battery Storage",
    "Grid": "Power Grid",
    "Homes": "Residential",
    "Industry": "Industrial",
  ),
  categories: (
    "Solar": "source",
    "Wind": "source",
    "Battery": "storage",
    "Grid": "distribution",
    "Homes": "consumption",
    "Industry": "consumption",
  ),
  tinter: tinter.categorical-tinter(
    palette: (green, yellow, blue, purple)
  ),
  ribbon-stylizer: ribbon-stylizer.gradient-from-to(
    transparency: 60%,
    stroke-width: 0.5pt,
  )
)
```



4.3 User Journey

Website user flow funnel.

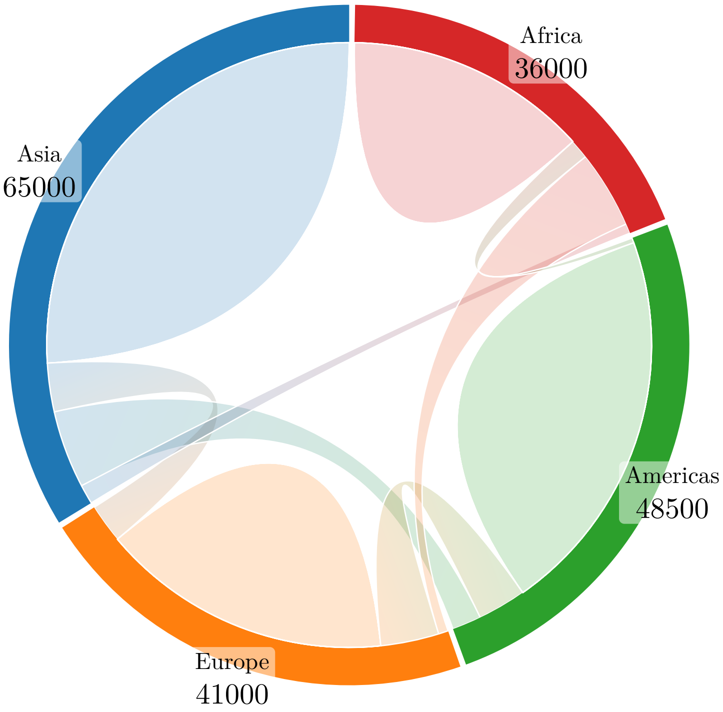
```
#sankey-diagram(
  (
    "Visitors": ("Homepage": 10000),
    "Homepage": ("Product Page": 5000, "Blog": 3000, "Exit": 2000),
    "Product Page": ("Cart": 2000, "Exit": 3000),
    "Blog": ("Product Page": 1000, "Exit": 2000),
    "Cart": ("Checkout": 1500, "Exit": 500),
    "Checkout": ("Purchase": 1200, "Exit": 300),
  ),
  tinter: tinter.layer-tinter(
    palette: (blue, cyan, green, yellow, orange, red)
  )
)
```



4.4 Trade Flows

International trade between regions.

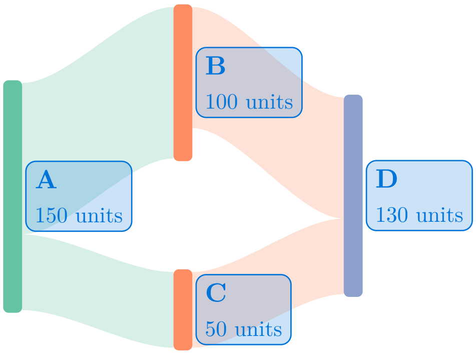
```
#chord-diagram(
  (
    "Asia": ("Asia": 50000, "Europe": 5000, "Americas": 8000, "Africa": 2000),
    "Europe": ("Asia": 4000, "Europe": 30000, "Americas": 6000, "Africa": 1000),
    "Americas": ("Asia": 3000, "Europe": 5000, "Americas": 40000, "Africa": 500),
    "Africa": ("Asia": 1000, "Europe": 8000, "Americas": 2000, "Africa": 25000),
  ),
  tinter: tinter.node-tinter(
    palette: palette.tableau
  ),
  ribbon-stylizer: ribbon-stylizer.gradient-from-to(
    transparency: 80%,
    stroke-width: 0.5pt,
    stroke-color: white,
  )
)
```



5 Advanced Styling

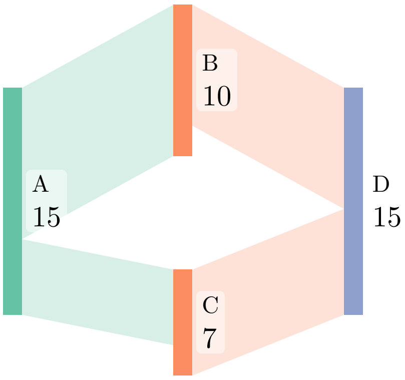
5.1 Custom Labels

```
#sankey-diagram(
  (
    "A": ("B": 100, "C": 50),
    "B": ("D": 80),
    "C": ("D": 50),
  ),
  draw-label: label.default-linear-label-drawer(
    styles: (
      inset: 0.3em,
      fill: blue.transparentize(80%),
      stroke: blue + 0.5pt,
      radius: 4pt,
    ),
    draw-content: (properties) => [
      #set text(fill: blue)
      #text(properties.name, size: 0.9em, weight: "bold") \
      #text(str(properties.size) + " units", size: 0.75em)
    ]
  )
)
```



5.2 Sharp Corners

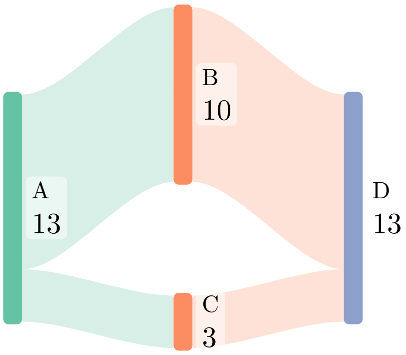
```
#sankey-diagram(
  (
    "A": ("B": 10, "C": 5),
    "B": ("D": 8),
    "C": ("D": 7),
  ),
  layout: layout.auto-linear(
    radius: 0pt,
    curve-factor: 0,
  )
)
```



5.3 Multiple Edges

Showing multiple connections between same nodes.

```
#sankey-diagram((
  ("A", "B", 2),
  ("A", "B", 3),
  ("A", "B", 5),
  ("A", "C", 3),
  ("B", "D", 10),
  ("C", "D", 3),
))
```



6 Data Format Examples

6.1 Adjacency Dictionary

```
#sankey-diagram((
  "A": ("B": 10, "C": 5),
  "B": ("D": 8),
  "C": ("D": 7),
))
```

6.2 Adjacency List

```
#sankey-diagram((
  ("A", "B", 10),
  ("A", "C", 5),
  ("B", "D", 8),
  ("C", "D", 7),
))
```

6.3 Adjacency Matrix

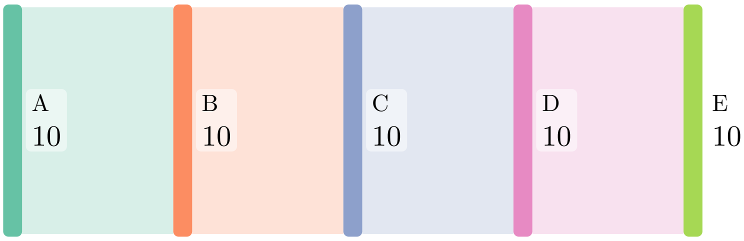
```
#chord-diagram((
  matrix: (
    (100, 50, 30),
    (50, 80, 40),
    (30, 40, 60),
  ),
  ids: ("X", "Y", "Z")
))
```

All three produce equivalent diagrams (shown above).

7 Color Palette Showcase

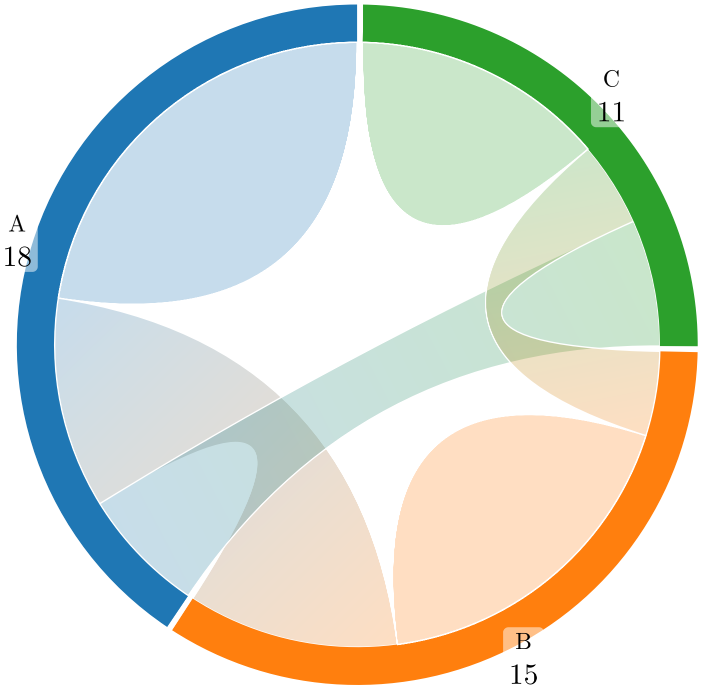
7.1 ColorBrewer Palette

```
#sankey-diagram(
  (
    "A": ("B": 10),
    "B": ("C": 10),
    "C": ("D": 10),
    "D": ("E": 10),
  ),
  tinter: tinter.layer-tinter(
    palette: palette.color-brewer-palette
  )
)
```



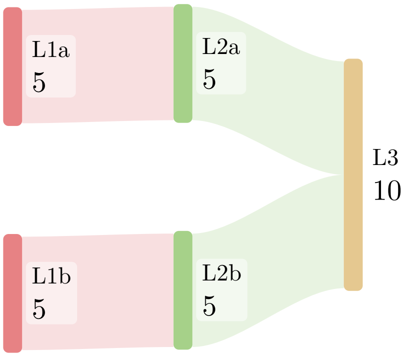
7.2 Tableau Palette

```
#chord-diagram(
  (
    "A": ("A": 10, "B": 5, "C": 3),
    "B": ("A": 5, "B": 8, "C": 2),
    "C": ("A": 3, "B": 2, "C": 6),
  ),
  tinter: tinter.node-tinter(
    palette: palette.tableau
  )
)
```



7.3 Catppuccin Palette

```
#sankey-diagram(
  (
    "L1a": ("L2a": 5),
    "L1b": ("L2b": 5),
    "L2a": ("L3": 5),
    "L2b": ("L3": 5),
  ),
  tinter: tinter.layer-tinter(
    palette: palette.catppuccin
  )
)
```



End of Examples Gallery

See `api-reference.typ` for complete API documentation