

1 柱状图数据堆叠

In [160]:

```
from pyecharts import Bar

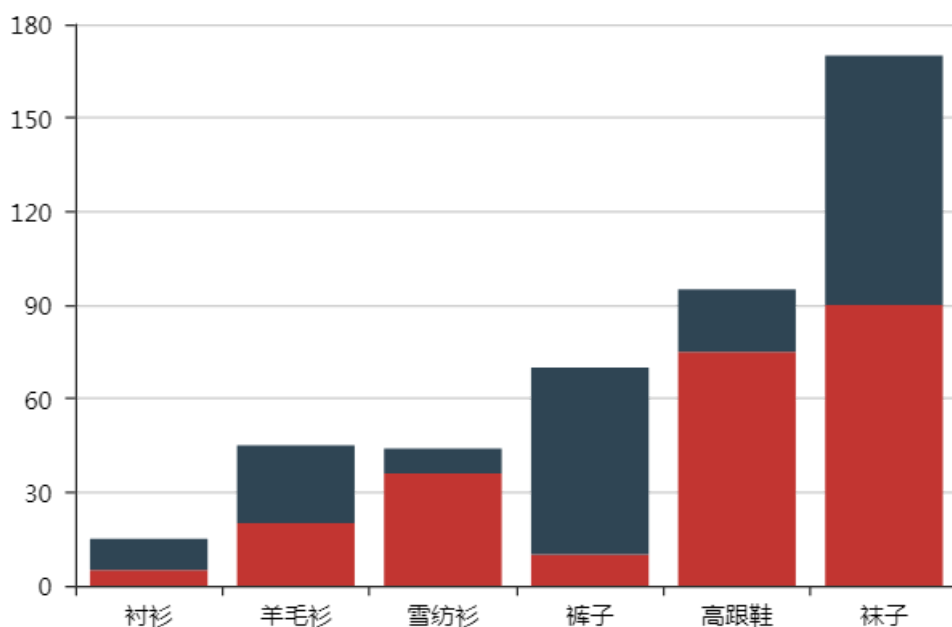
attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
v1 = [5, 20, 36, 10, 75, 90]
v2 = [10, 25, 8, 60, 20, 80]
bar = Bar("柱状图数据堆叠示例", width=550)
bar.add("商家A", attr, v1, is_stack=True)
bar.add("商家B", attr, v2, is_stack=True)

bar
```

Out[160]:

柱状图数据堆叠示例

商家A 商家B



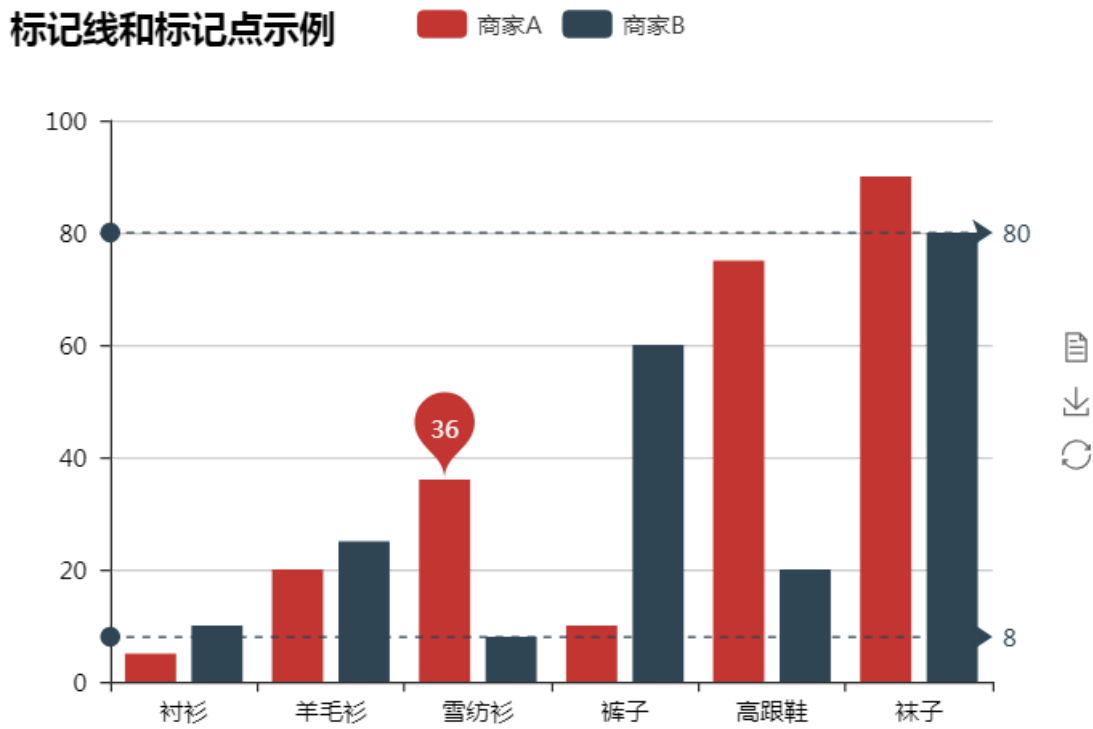
2 标记线和标记点示例

In [161]:

```
bar = Bar("标记线和标记点示例", width=550)
bar.add("商家A", attr, v1, mark_point=["average"])
bar.add("商家B", attr, v2, mark_line=["min", "max"])

bar
```

Out[161]:



3 x轴和y轴交换

In [162]:

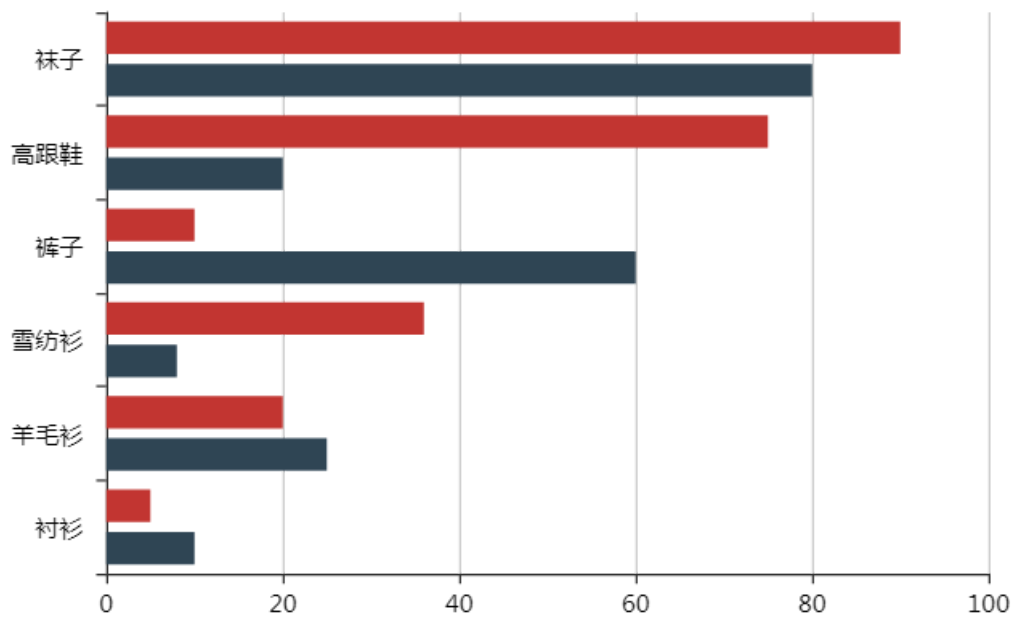
```
bar = Bar("x 轴和 y 轴交换", width=550)
bar.add("商家A", attr, v1)
bar.add("商家B", attr, v2, is_convert=True)
```

bar

Out[162]:

x 轴和 y 轴交换

商家A 商家B



4 滑块数据缩放

In [163]:

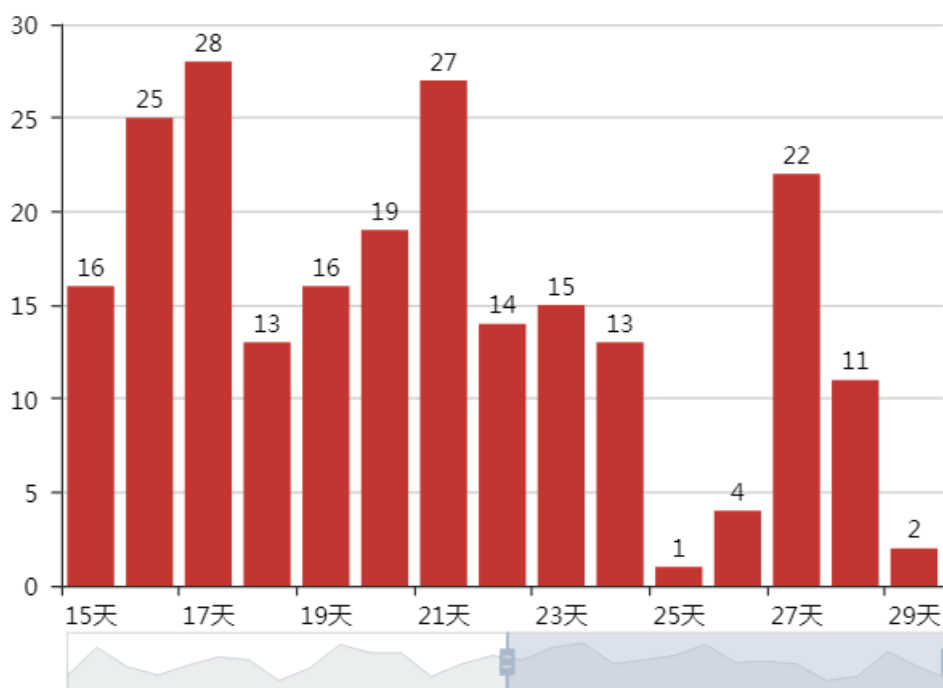
```
import random

attr = ["{}天".format(i) for i in range(30)]
v1 = [random.randint(1, 30) for _ in range(30)]
bar = Bar("Bar - datazoom - slider 示例", width=550)
bar.add("", attr, v1, is_label_show=True, is_datazoom_show=True)

bar
```

Out[163]:

Bar - datazoom - slider 示例



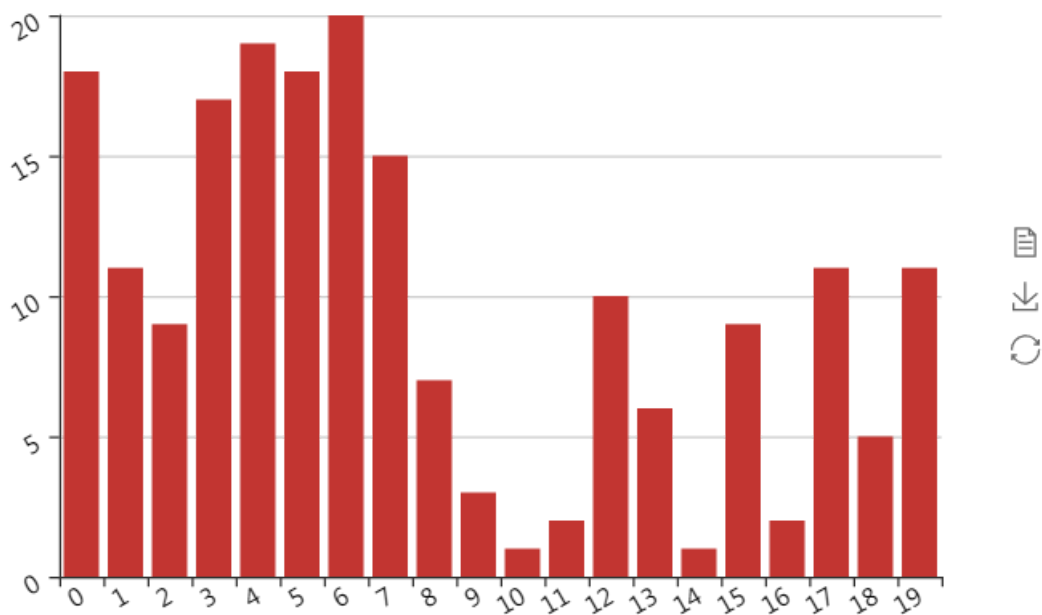
5 坐标轴标签旋转示例

In [164]:

```
attr = ["{}".format(i) for i in range(20)]  
v1 = [random.randint(1, 20) for _ in range(20)]  
bar = Bar("坐标轴标签旋转示例", width=550)  
bar.add("", attr, v1, xaxis_interval=0, xaxis_rotate=30, yaxis_rotate=30)  
  
bar
```

Out[164]:

坐标轴标签旋转示例



6 3D柱状图示例

In [165]:

```

from pyecharts import Bar3D

bar3d = Bar3D("3D柱状图示例", width=550)
x_axis = ["12a", "1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a", "10a", "11a",
          "12p", "1p", "2p", "3p", "4p", "5p", "6p", "7p", "8p", "9p", "10p", "11p"]
y_axis = ["Saturday", "Friday", "Thursday", "Wednesday", "Tuesday", "Monday", "Sunday"]
data = [[0, 0, 5], [0, 1, 1], [0, 2, 0], [0, 3, 0], [0, 4, 0], [0, 5, 0], [0, 6, 0], [0, 7, 0],
        [0, 8, 0], [0, 9, 0], [0, 10, 0], [0, 11, 2], [0, 12, 4], [0, 13, 1], [0, 14, 1], [0, 15, 3],
        [0, 16, 4], [0, 17, 6], [0, 18, 4], [0, 19, 4], [0, 20, 3], [0, 21, 3], [0, 22, 2], [0, 23,
        [1, 0, 7], [1, 1, 0], [1, 2, 0], [1, 3, 0], [1, 4, 0], [1, 5, 0], [1, 6, 0], [1, 7, 0], [1,
        [1, 9, 0], [1, 10, 5], [1, 11, 2], [1, 12, 2], [1, 13, 6], [1, 14, 9], [1, 15, 11], [1, 16,
        [1, 18, 8], [1, 19, 12], [1, 20, 5], [1, 21, 5], [1, 22, 7], [1, 23, 2], [2, 0, 1], [2, 1, 1
        [2, 2, 0], [2, 3, 0], [2, 4, 0], [2, 5, 0], [2, 6, 0], [2, 7, 0], [2, 8, 0], [2, 9, 0], [2,
        [2, 11, 2], [2, 12, 1], [2, 13, 9], [2, 14, 8], [2, 15, 10], [2, 16, 6], [2, 17, 5], [2, 18,
        [2, 19, 5], [2, 20, 7], [2, 21, 4], [2, 22, 2], [2, 23, 4], [3, 0, 7], [3, 1, 3], [3, 2, 0],
        [3, 4, 0], [3, 5, 0], [3, 6, 0], [3, 7, 0], [3, 8, 1], [3, 9, 0], [3, 10, 5], [3, 11, 4], [3
        [3, 13, 14], [3, 14, 13], [3, 15, 12], [3, 16, 9], [3, 17, 5], [3, 18, 5], [3, 19, 10], [3,
        [3, 21, 4], [3, 22, 4], [3, 23, 1], [4, 0, 1], [4, 1, 3], [4, 2, 0], [4, 3, 0], [4, 4, 0], [
        [4, 6, 0], [4, 7, 0], [4, 8, 0], [4, 9, 2], [4, 10, 4], [4, 11, 4], [4, 12, 2], [4, 13, 4],
        [4, 15, 14], [4, 16, 12], [4, 17, 1], [4, 18, 8], [4, 19, 5], [4, 20, 3], [4, 21, 7], [4, 22
        [4, 23, 0], [5, 0, 2], [5, 1, 1], [5, 2, 0], [5, 3, 3], [5, 4, 0], [5, 5, 0], [5, 6, 0], [5,
        [5, 8, 2], [5, 9, 0], [5, 10, 4], [5, 11, 1], [5, 12, 5], [5, 13, 10], [5, 14, 5], [5, 15, 7
        [5, 17, 6], [5, 18, 0], [5, 19, 5], [5, 20, 3], [5, 21, 4], [5, 22, 2], [5, 23, 0], [6, 0, 1
        [6, 2, 0], [6, 3, 0], [6, 4, 0], [6, 5, 0], [6, 6, 0], [6, 7, 0], [6, 8, 0], [6, 9, 0], [6,
        [6, 11, 0], [6, 12, 2], [6, 13, 1], [6, 14, 3], [6, 15, 4], [6, 16, 0], [6, 17, 0], [6, 18,
        [6, 20, 1], [6, 21, 2], [6, 22, 2], [6, 23, 6]]
range_color = ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf',
               '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026']
bar3d.add("", x_axis, y_axis, [[d[1], d[0], d[2]] for d in data], is_visualmap=True,
          visual_range=[0, 20], visual_range_color=range_color, grid3d_width=200, grid3d_depth=80)

bar3d

```

Out[165]:

3D柱状图示例



7 动态散点图各种图形示例

In [166]:

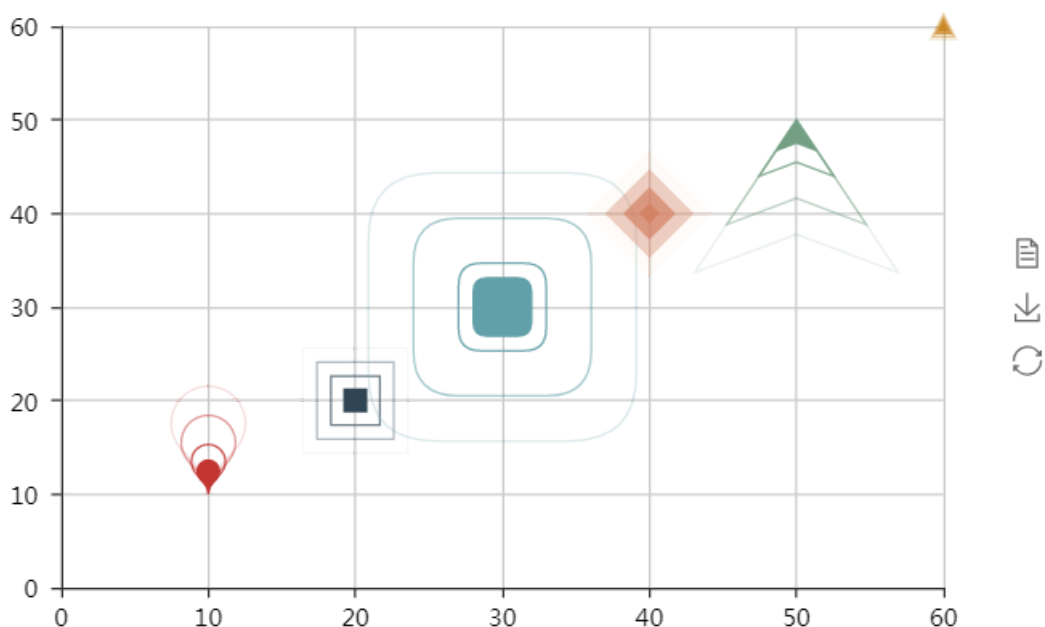
```
from pyecharts import EffectScatter

es = EffectScatter("动态散点图各种图形示例", width=550)
es.add("", [10], [10], symbol_size=20, effect_scale=3.5, effect_period=3, symbol="pin")
es.add("", [20], [20], symbol_size=12, effect_scale=4.5, effect_period=4, symbol="rect")
es.add("", [30], [30], symbol_size=30, effect_scale=5.5, effect_period=5, symbol="roundRect")
es.add("", [40], [40], symbol_size=10, effect_scale=6.5, effect_brushtype='fill', symbol="diamond")
es.add("", [50], [50], symbol_size=16, effect_scale=5.5, effect_period=3, symbol="arrow")
es.add("", [60], [60], symbol_size=6, effect_scale=2.5, effect_period=3, symbol="triangle")

es
```

Out[166]:

动态散点图各种图形示例



8 漏斗图示例

In [167]:

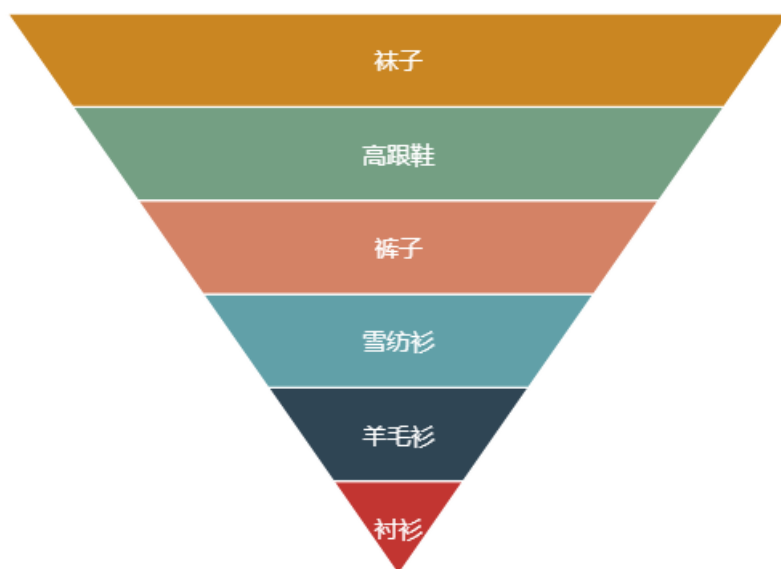
```
from pyecharts import Funnel

attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
value = [20, 40, 60, 80, 100, 120]
funnel = Funnel("漏斗图示例", width=550)
funnel.add("商品", attr, value, is_label_show=True, label_pos="inside", label_text_color="#fff")

funnel
```

Out[167]:

漏斗图示例 裤子 高跟鞋 羊毛衫 雪纺衫 衬衫 袜子



9 仪表盘示例

In [168]:

```
from pyecharts import Gauge

gauge = Gauge("仪表盘示例", width=550)
gauge.add("业务示例", "完成率", 66.66)

gauge
```

Out[168]:

仪表盘示例



10 地图

In [169]:

```

from pyecharts import Geo

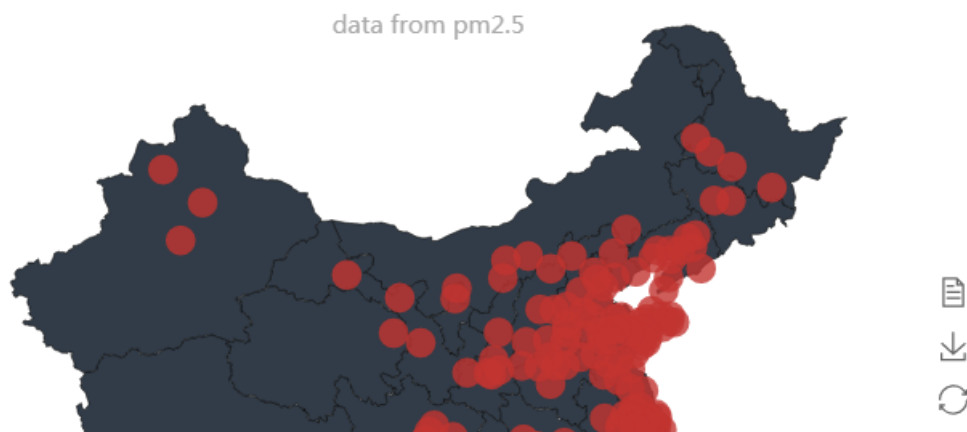
data = [
    ("海门", 9), ("鄂尔多斯", 12), ("招远", 12), ("舟山", 12), ("齐齐哈尔", 14), ("盐城", 15),
    ("赤峰", 16), ("青岛", 18), ("乳山", 18), ("金昌", 19), ("泉州", 21), ("莱西", 21),
    ("日照", 21), ("胶南", 22), ("南通", 23), ("拉萨", 24), ("云浮", 24), ("梅州", 25),
    ("文登", 25), ("上海", 25), ("攀枝花", 25), ("威海", 25), ("承德", 25), ("厦门", 26),
    ("汕尾", 26), ("潮州", 26), ("丹东", 27), ("太仓", 27), ("曲靖", 27), ("烟台", 28),
    ("福州", 29), ("瓦房店", 30), ("即墨", 30), ("抚顺", 31), ("玉溪", 31), ("张家口", 31),
    ("阳泉", 31), ("莱州", 32), ("湖州", 32), ("汕头", 32), ("昆山", 33), ("宁波", 33),
    ("湛江", 33), ("揭阳", 34), ("荣成", 34), ("连云港", 35), ("葫芦岛", 35), ("常熟", 36),
    ("东莞", 36), ("河源", 36), ("淮安", 36), ("泰州", 36), ("南宁", 37), ("营口", 37),
    ("惠州", 37), ("江阴", 37), ("蓬莱", 37), ("韶关", 38), ("嘉峪关", 38), ("广州", 38),
    ("延安", 38), ("太原", 39), ("清远", 39), ("中山", 39), ("昆明", 39), ("寿光", 40),
    ("盘锦", 40), ("长治", 41), ("深圳", 41), ("珠海", 42), ("宿迁", 43), ("咸阳", 43),
    ("铜川", 44), ("平度", 44), ("佛山", 44), ("海口", 44), ("江门", 45), ("章丘", 45),
    ("肇庆", 46), ("大连", 47), ("临汾", 47), ("吴江", 47), ("石嘴山", 49), ("沈阳", 50),
    ("苏州", 50), ("茂名", 50), ("嘉兴", 51), ("长春", 51), ("胶州", 52), ("银川", 52),
    ("张家港", 52), ("三门峡", 53), ("锦州", 54), ("南昌", 54), ("柳州", 54), ("三亚", 54),
    ("自贡", 56), ("吉林", 56), ("阳江", 57), ("泸州", 57), ("西宁", 57), ("宜宾", 58),
    ("呼和浩特", 58), ("成都", 58), ("大同", 58), ("镇江", 59), ("桂林", 59), ("张家界", 59),
    ("宜兴", 59), ("北海", 60), ("西安", 61), ("金坛", 62), ("东营", 62), ("牡丹江", 63),
    ("遵义", 63), ("绍兴", 63), ("扬州", 64), ("常州", 64), ("潍坊", 65), ("重庆", 66),
    ("台州", 67), ("南京", 67), ("滨州", 70), ("贵阳", 71), ("无锡", 71), ("本溪", 71),
    ("克拉玛依", 72), ("渭南", 72), ("马鞍山", 72), ("宝鸡", 72), ("焦作", 75), ("句容", 75),
    ("北京", 79), ("徐州", 79), ("衡水", 80), ("包头", 80), ("绵阳", 80), ("乌鲁木齐", 84),
    ("枣庄", 84), ("杭州", 84), ("淄博", 85), ("鞍山", 86), ("溧阳", 86), ("库尔勒", 86),
    ("安阳", 90), ("开封", 90), ("济南", 92), ("德阳", 93), ("温州", 95), ("九江", 96),
    ("邯郸", 98), ("临安", 99), ("兰州", 99), ("沧州", 100), ("临沂", 103), ("南充", 104),
    ("天津", 105), ("富阳", 106), ("泰安", 112), ("诸暨", 112), ("郑州", 113), ("哈尔滨", 114),
    ("聊城", 116), ("芜湖", 117), ("唐山", 119), ("平顶山", 119), ("邢台", 119), ("德州", 120),
    ("济宁", 120), ("荆州", 127), ("宜昌", 130), ("义乌", 132), ("丽水", 133), ("洛阳", 134),
    ("秦皇岛", 136), ("株洲", 143), ("石家庄", 147), ("莱芜", 148), ("常德", 152), ("保定", 153),
    ("湘潭", 154), ("金华", 157), ("岳阳", 169), ("长沙", 175), ("衢州", 177), ("廊坊", 193),
    ("菏泽", 194), ("合肥", 229), ("武汉", 273), ("大庆", 279)]

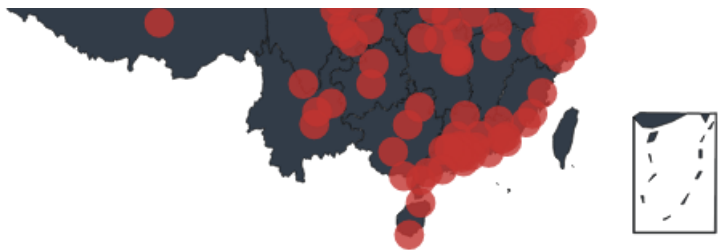
geo = Geo("全国主要城市空气质量", "data from pm2.5", title_color="#fff", title_pos="center",
          width=550, background_color="#404a59")
attr, value = geo.cast(data)
geo.add("", attr, value, visual_range=[0, 200], visual_text_color="#fff", symbol_size=15,)

geo

```

Out[169]:



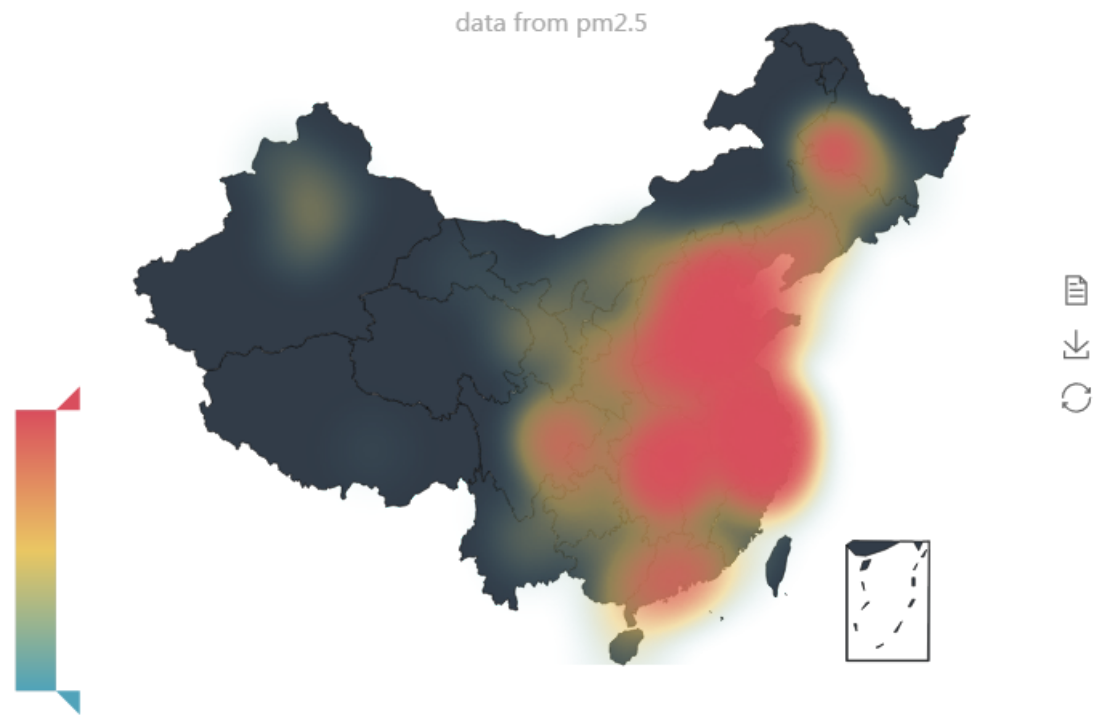


11 热力图

In [170]:

```
geo = Geo("全国主要城市空气质量", "data from pm2.5", title_color="#fff", title_pos="center",
          width = 550, background_color="#404a59")
geo.add("", attr, value, type="heatmap", is_visualmap=True, visual_range=[0, 300], visual_text_color="white")
attr, value = geo.cast(data)
geo
```

Out[170]:



12 引力关系图示例

In [171]:

```
from pyecharts import Graph

nodes = [{"name": "结点1", "symbolSize": 10},
         {"name": "结点2", "symbolSize": 20},
         {"name": "结点3", "symbolSize": 30},
         {"name": "结点4", "symbolSize": 40},
         {"name": "结点5", "symbolSize": 50},
         {"name": "结点6", "symbolSize": 40},
         {"name": "结点7", "symbolSize": 30},
         {"name": "结点8", "symbolSize": 20}]

links = []
for i in nodes:
    for j in nodes:
        links.append({"source": i.get('name'), "target": j.get('name')})
graph = Graph("关系图-力引导布局示例", width=550)
graph.add("", nodes, links, repulsion=8000)

graph
```

Out[171]:

关系图-力引导布局示例



13 热力图直角坐标系

In [172]:

```

from pyecharts import HeatMap

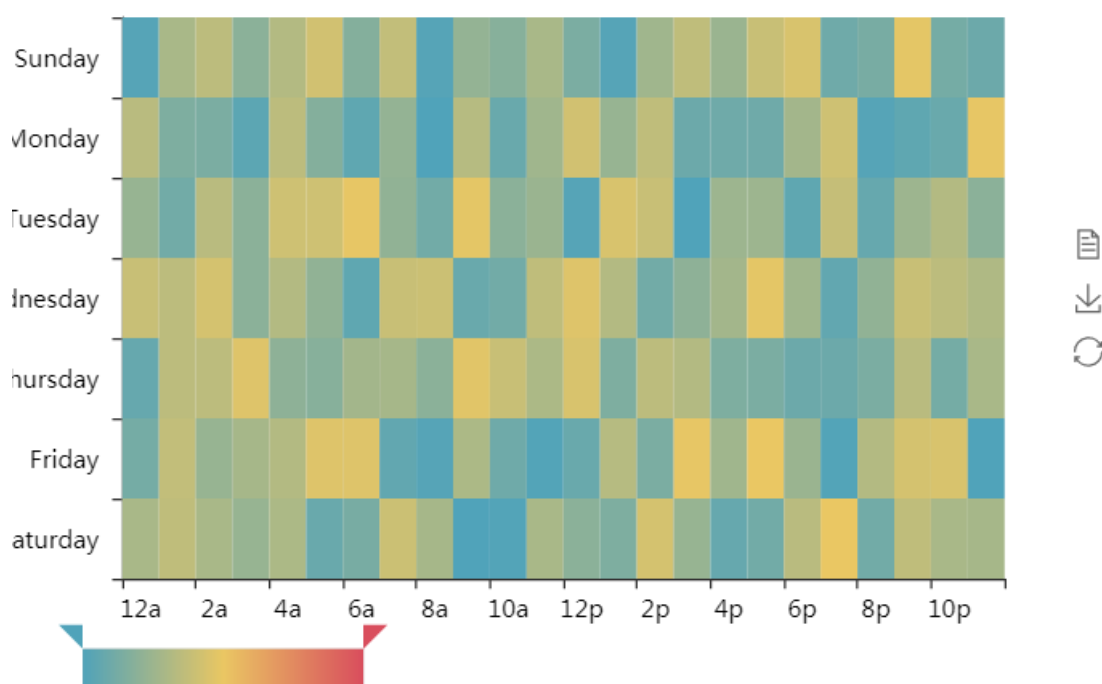
x_axis = ["12a", "1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a", "10a", "11a",
          "12p", "1p", "2p", "3p", "4p", "5p", "6p", "7p", "8p", "9p", "10p", "11p"]
y_axis = ["Saturday", "Friday", "Thursday", "Wednesday", "Tuesday", "Monday", "Sunday"]
data = [[i, j, random.randint(0, 50)] for i in range(24) for j in range(7)]
heatmap = HeatMap("热力图直角坐标系", width=550)
heatmap.add("", x_axis, y_axis, data, is_visualmap=True, visual_text_color="#fff",
            visual_orient="horizontal")

heatmap

```

Out[172]:

热力图直角坐标系



14 K线图

In [173]:

```

from pyecharts import Kline

v = [[2320.26, 2320.26, 2287.3, 2362.94], [2300, 2291.3, 2288.26, 2308.38],
      [2295.35, 2346.5, 2295.35, 2345.92], [2347.22, 2358.98, 2337.35, 2363.8],
      [2360.75, 2382.48, 2347.89, 2383.76], [2383.43, 2385.42, 2371.23, 2391.82],
      [2377.41, 2419.02, 2369.57, 2421.15], [2425.92, 2428.15, 2417.58, 2440.38],
      [2411, 2433.13, 2403.3, 2437.42], [2432.68, 2334.48, 2427.7, 2441.73],
      [2430.69, 2418.53, 2394.22, 2433.89], [2416.62, 2432.4, 2414.4, 2443.03],
      [2441.91, 2421.56, 2418.43, 2444.8], [2420.26, 2382.91, 2373.53, 2427.07],
      [2383.49, 2397.18, 2370.61, 2397.94], [2378.82, 2325.95, 2309.17, 2378.82],
      [2322.94, 2314.16, 2308.76, 2330.88], [2320.62, 2325.82, 2315.01, 2338.78],
      [2313.74, 2293.34, 2289.89, 2340.71], [2297.77, 2313.22, 2292.03, 2324.63],
      [2322.32, 2365.59, 2308.92, 2366.16], [2364.54, 2359.51, 2330.86, 2369.65],
      [2332.08, 2273.4, 2259.25, 2333.54], [2274.81, 2326.31, 2270.1, 2328.14],
      [2333.61, 2347.18, 2321.6, 2351.44], [2340.44, 2324.29, 2304.27, 2352.02],
      [2326.42, 2318.61, 2314.59, 2333.67], [2314.68, 2310.59, 2296.58, 2320.96],
      [2309.16, 2286.6, 2264.83, 2333.29], [2282.17, 2263.97, 2253.25, 2286.33],
      [2255.77, 2270.28, 2253.31, 2276.22]]

kline = Kline("K线图示例", width=550)
kline.add("日K", ["2018/1/{0}".format(i + 1) for i in range(31)], v)

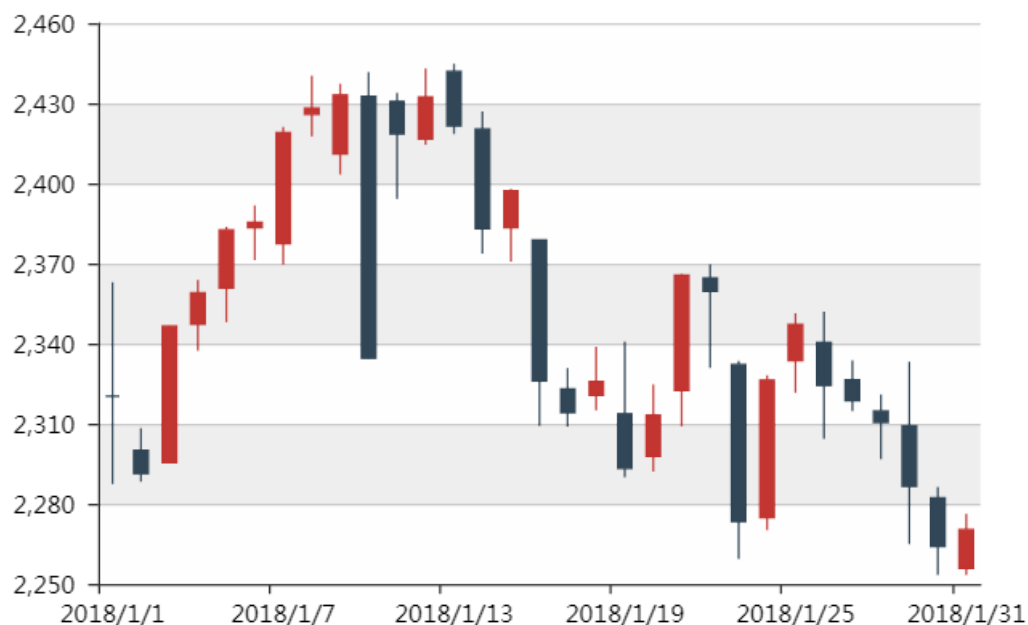
kline

```

Out[173]:

K线图示例

日K



15 折线图示例

In [174]:

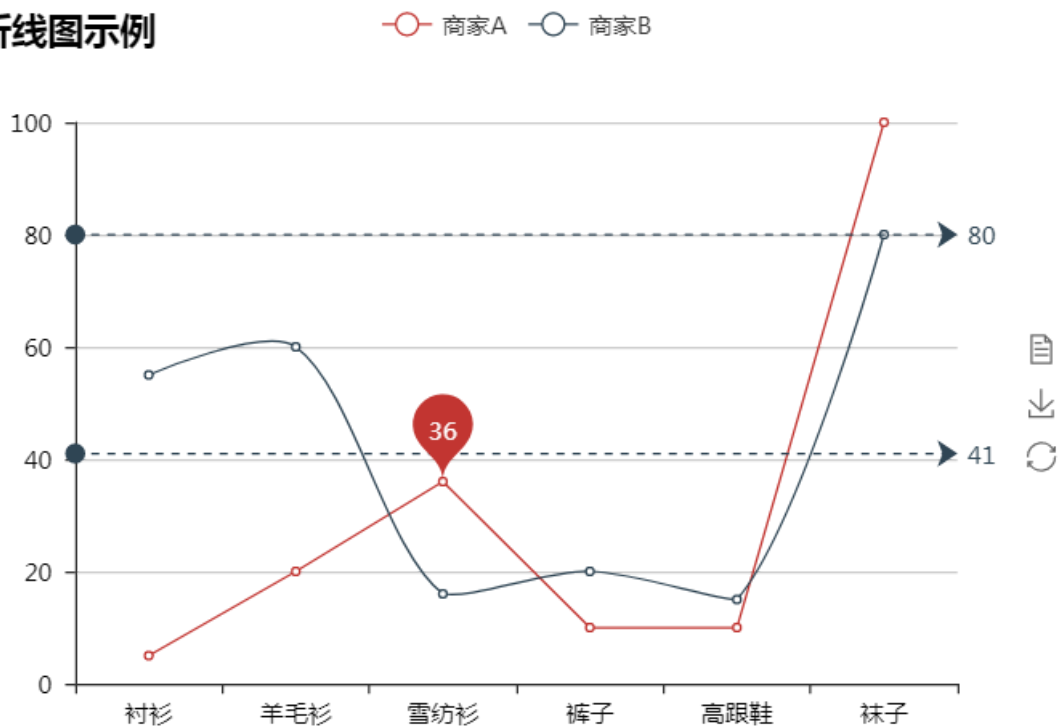
```
from pyecharts import Line

attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
v1 = [5, 20, 36, 10, 10, 100]
v2 = [55, 60, 16, 20, 15, 80]
line = Line("折线图示例", width=550)
line.add("商家A", attr, v1, mark_point=["average"])
line.add("商家B", attr, v2, is_smooth=True, mark_line=["max", "average"])

line
```

Out[174]:

折线图示例



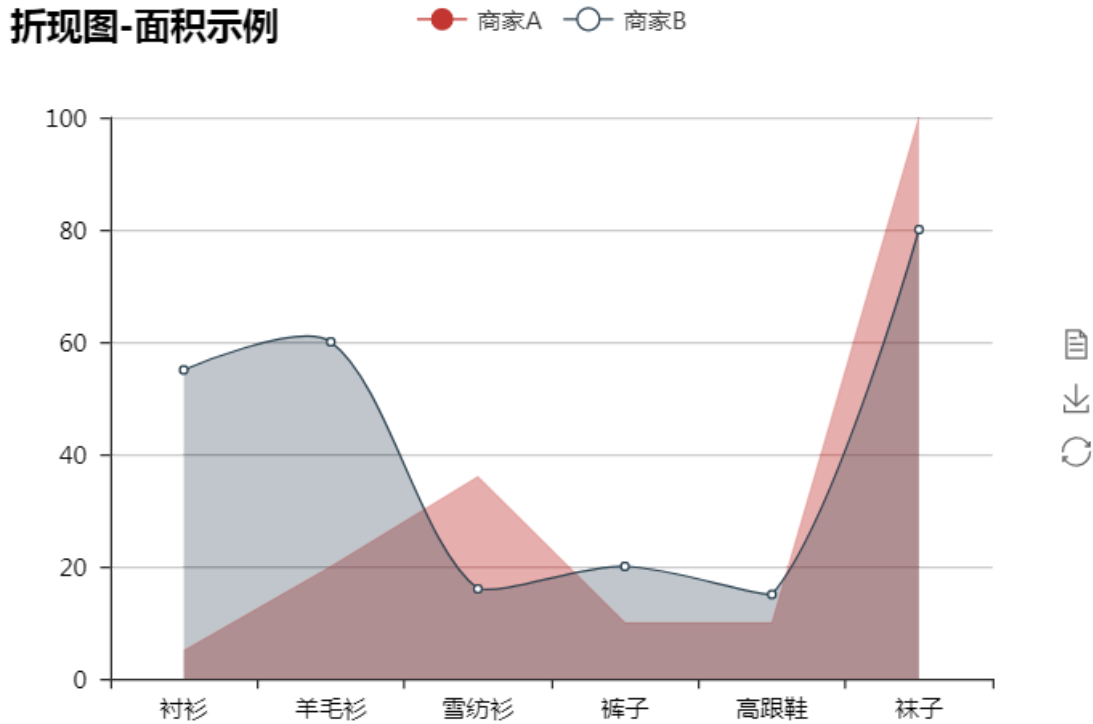
16 折线图-面积示例图

In [175]:

```
line = Line("折线图-面积示例", width=550)
line.add("商家A", attr, v1, is_fill=True, line_opacity=0.2, area_opacity=0.4, symbol=None)
line.add("商家B", attr, v2, is_fill=True, line_opacity="#000", area_opacity=0.3, is_smooth=True)

line
```

Out[175]:



17 3D折现示例图

In [176]:

```

from pyecharts import Line3D

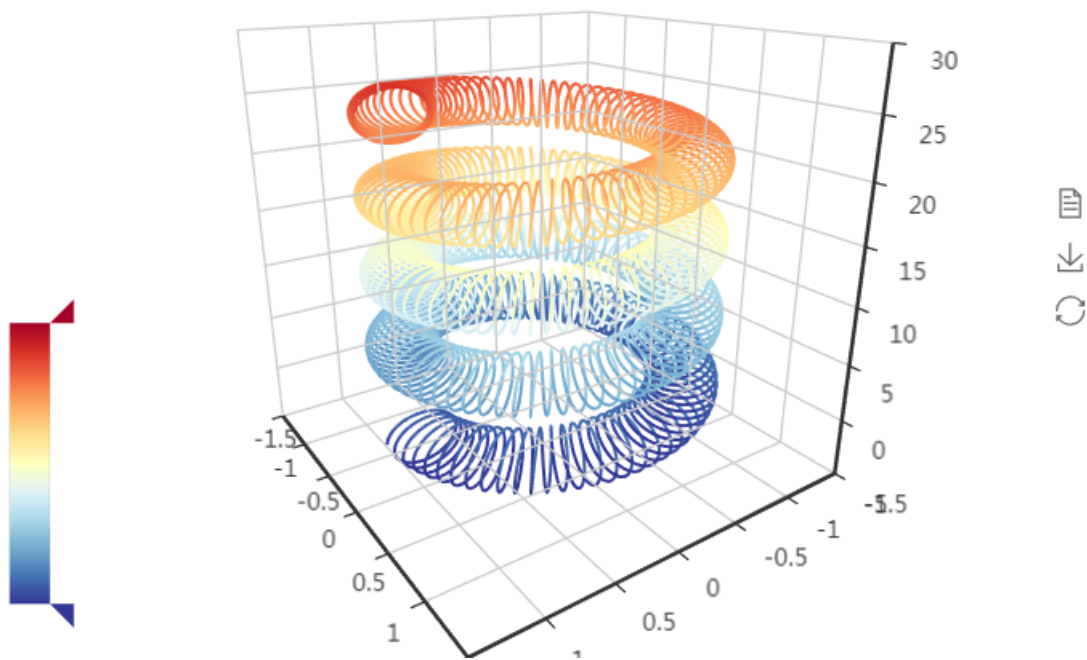
import math
_data = []
for t in range(0, 25000):
    _t = t/1000
    x = (1 + 0.25 * math.cos(75 * _t)) * math.cos(_t)
    y = (1 + 0.25 * math.cos(75 * _t)) * math.sin(_t)
    z = _t + 2.0 * math.sin(75 * _t)
    _data.append([x, y, z])
range_color = ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf',
               '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026']
line3d = Line3D("3D折线图示例", width=550)
line3d.add("", _data, is_visualmap=True, visual_range_color=range_color, visual_range=[0, 30],
           is_grid3d_rotate=True, grid3d_rotate_speed=100)

line3d

```

Out[176]:

3D折线图示例



18 水球图示例

In [177]:

```
from pyecharts import Liquid

liquid = Liquid("水球图示例", width=550)
liquid.add("Liquid", [0.6, 0.5, 0.4, 0.3], is_liquid_outline_show=False)

liquid
```

Out[177]:

水球图示例



19 全国地图示例

In [178]:

```
from pyecharts import Map

value = [155, 10, 66, 78]
attr = ["福建", "山东", "北京", "上海"]
map = Map("全国地图示例", width=550)
map.add("", attr, value, maptype="china")

map
```

Out[178]:

全国地图示例



20 广东地图示例

In [179]:

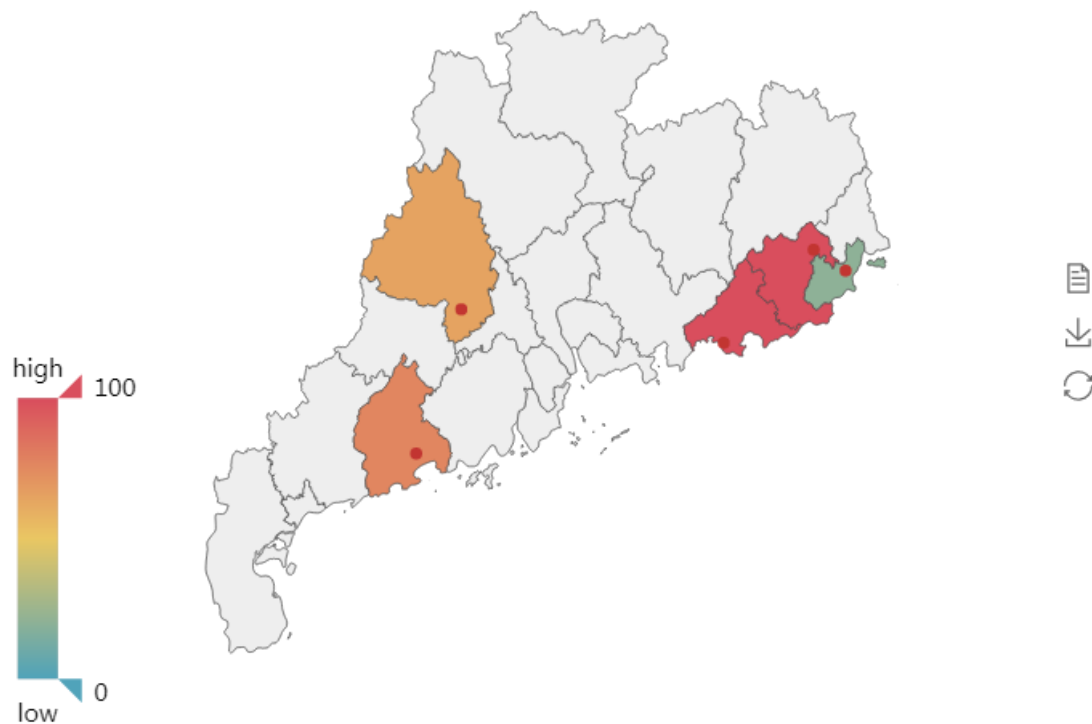
```
from pyecharts import Map

value = [20, 190, 253, 77, 65]
attr = ['汕头市', '汕尾市', '揭阳市', '阳江市', '肇庆市']
map = Map("广东地图示例", width=550)
map.add("", attr, value, maptype="广东", is_visualmap=True, visual_text_color="#000")

map
```

Out[179]:

广东地图示例



21 世界地图示例

In [180]:

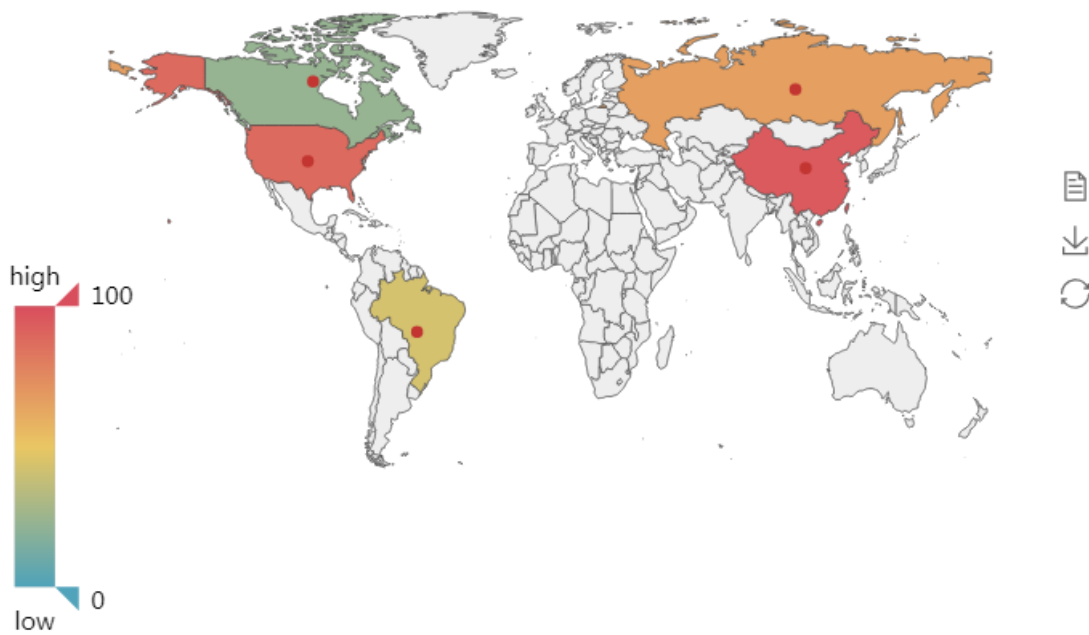
```
value = [95.1, 23.2, 43.3, 66.4, 88.5]
attr= ["China", "Canada", "Brazil", "Russia", "United States"]
map = Map("世界地图示例", width=550)

map.add("", attr, value, maptype="world", is_visualmap=True, visual_text_color="#000")

map
```

Out[180]:

世界地图示例



22 平行坐标系-默认指示器

In [181]:

```

from pyecharts import Parallel

schema = ["data", "AQI", "PM2.5", "PM10", "CO", "NO2"]
data = [
    [1, 91, 45, 125, 0.82, 34],
    [2, 65, 27, 78, 0.86, 45],
    [3, 83, 60, 84, 1.09, 73],
    [4, 109, 81, 121, 1.28, 68],
    [5, 106, 77, 114, 1.07, 55],
    [6, 109, 81, 121, 1.28, 68],
    [7, 106, 77, 114, 1.07, 55],
    [8, 89, 65, 78, 0.86, 51, 26],
    [9, 53, 33, 47, 0.64, 50, 17],
    [10, 80, 55, 80, 1.01, 75, 24],
    [11, 117, 81, 124, 1.03, 45]
]

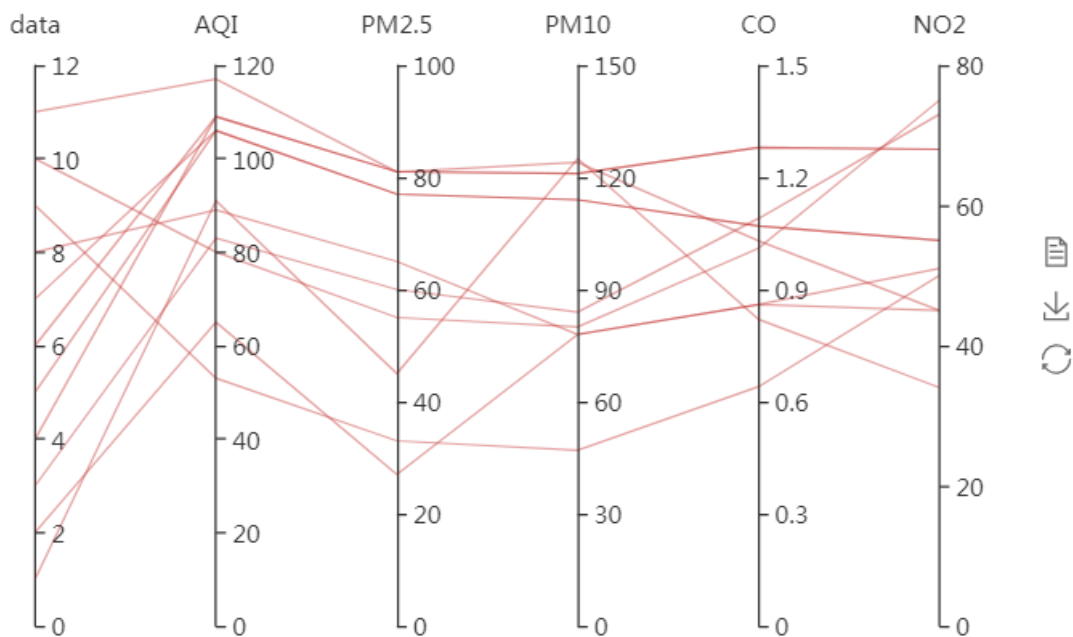
parallel = Parallel("平行坐标系-默认指示器", width=550)
parallel.config(schema)
parallel.add("parallel", data, is_random=True)

parallel

```

Out[181]:

平行坐标系-默认指示器

■ parallel


饼图示例

In [182]:

```
from pyecharts import Pie

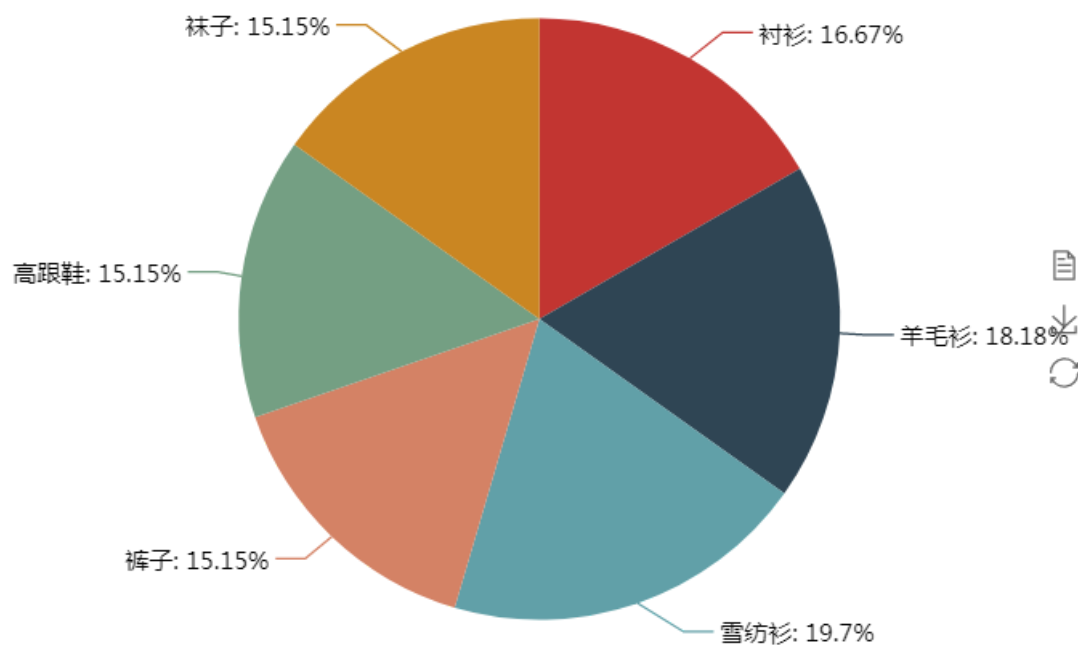
attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
v1 = [11, 12, 13, 10, 10, 10]
pie = Pie("饼图示例", width=550)
pie.add("", attr, v1, is_label_show=True)

pie
```

Out[182]:

饼图示例

类别	值	百分比
衬衫	11	16.67%
羊毛衫	12	18.18%
雪纺衫	13	19.7%
裤子	10	15.15%
高跟鞋	10	15.15%
袜子	10	15.15%



饼图-玫瑰花示例

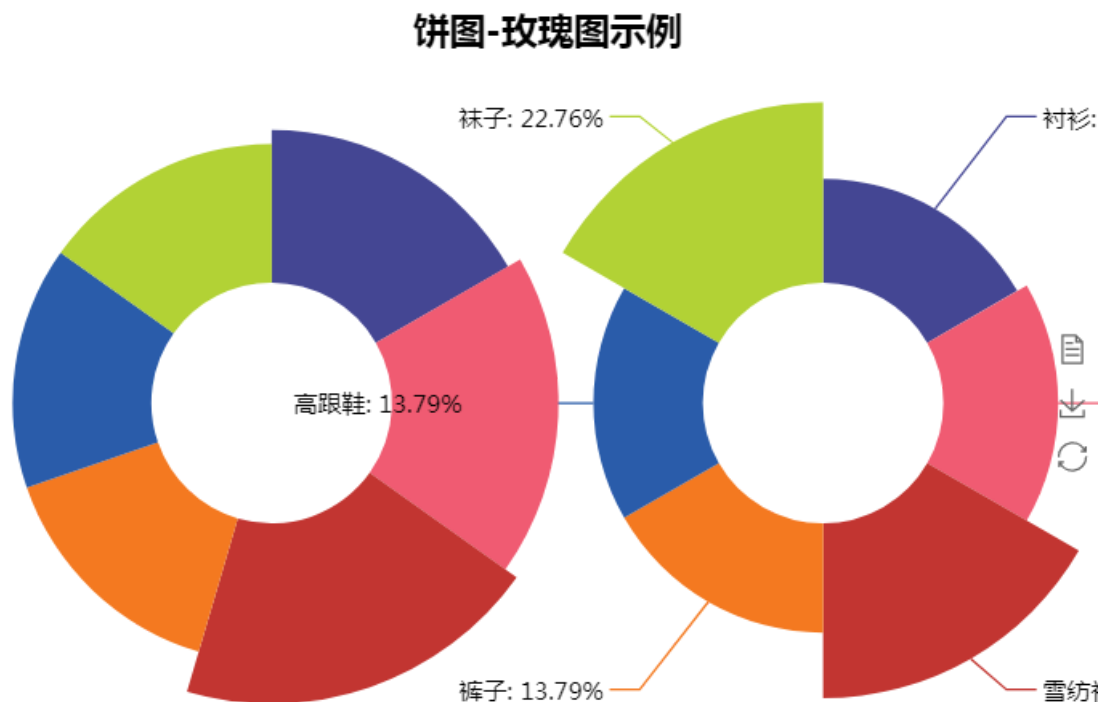
In [183]:

```
attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
v1 = [11, 12, 13, 10, 10, 10]
v2 = [19, 21, 32, 20, 20, 33]

pie = Pie("饼图-玫瑰图示例", title_pos="center", width=550)
pie.add("商户A", attr, v1, center=[25, 50], is_random=True, radius=[30, 75], rosetype="radius")
pie.add("商户B", attr, v2, center=[75, 50], is_random=True, radius=[30, 75], rosetype="area",
       is_legend_show=False, is_label_show=True)

pie
```

Out[183]:



极坐标散点图

In [185]:

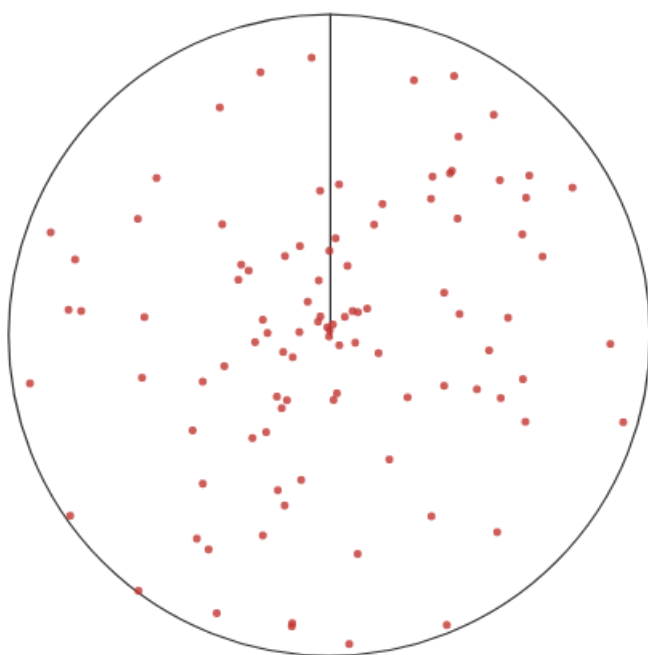
```
from pyecharts import Polar

import random
data = [(i, random.randint(1, 100)) for i in range(101)]
polar = Polar("极坐标系-散点图示例", width=550)
polar.add("", data, boundary_gap=False, type='scatter', is_splitline_show=False,
          area_color=None, is_axisline_show=True)

polar
```

Out[185]:

极坐标系-散点图示例



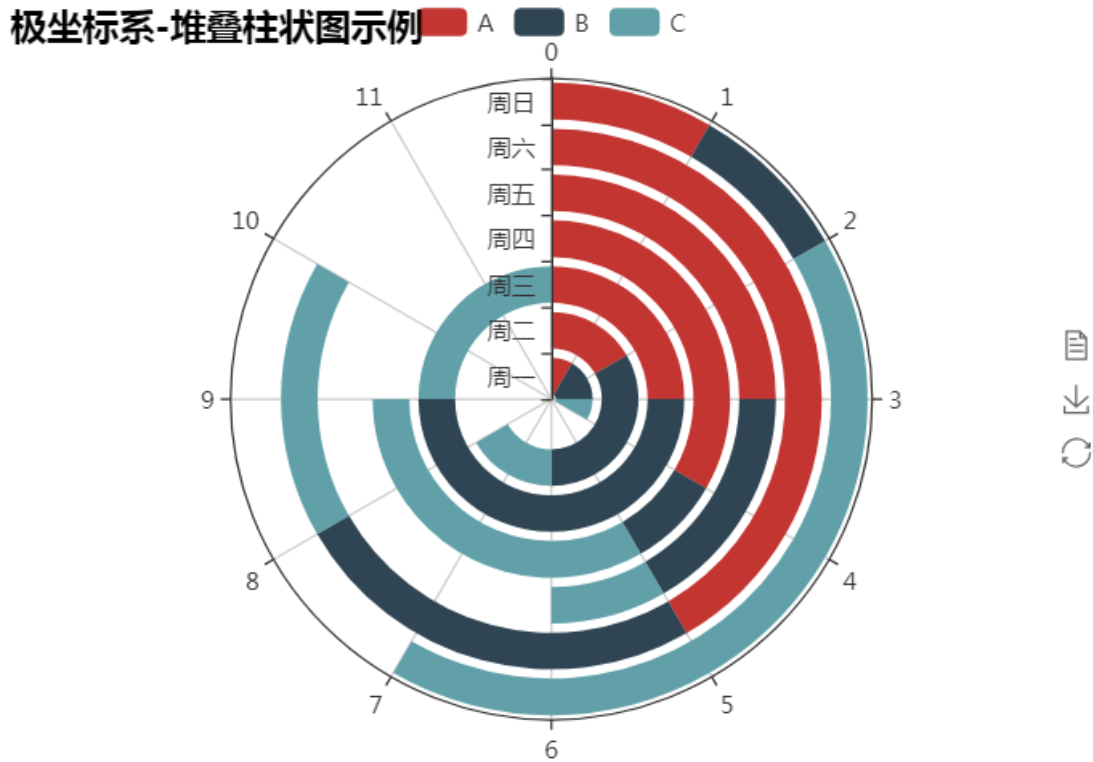
极坐标系-堆叠柱状图示例

In [186]:

```
radius = ['周一', '周二', '周三', '周四', '周五', '周六', '周日']
polar = Polar("极坐标系-堆叠柱状图示例", width=550)
polar.add("A", [1, 2, 3, 4, 3, 5, 1], radius_data=radius, type="barRadius", is_stack=True)
polar.add("B", [2, 4, 6, 1, 2, 3, 1], radius_data=radius, type="barRadius", is_stack=True)
polar.add("C", [1, 2, 3, 4, 1, 2, 5], radius_data=radius, type="barRadius", is_stack=True)

polar
```

Out[186]:



雷达图示例

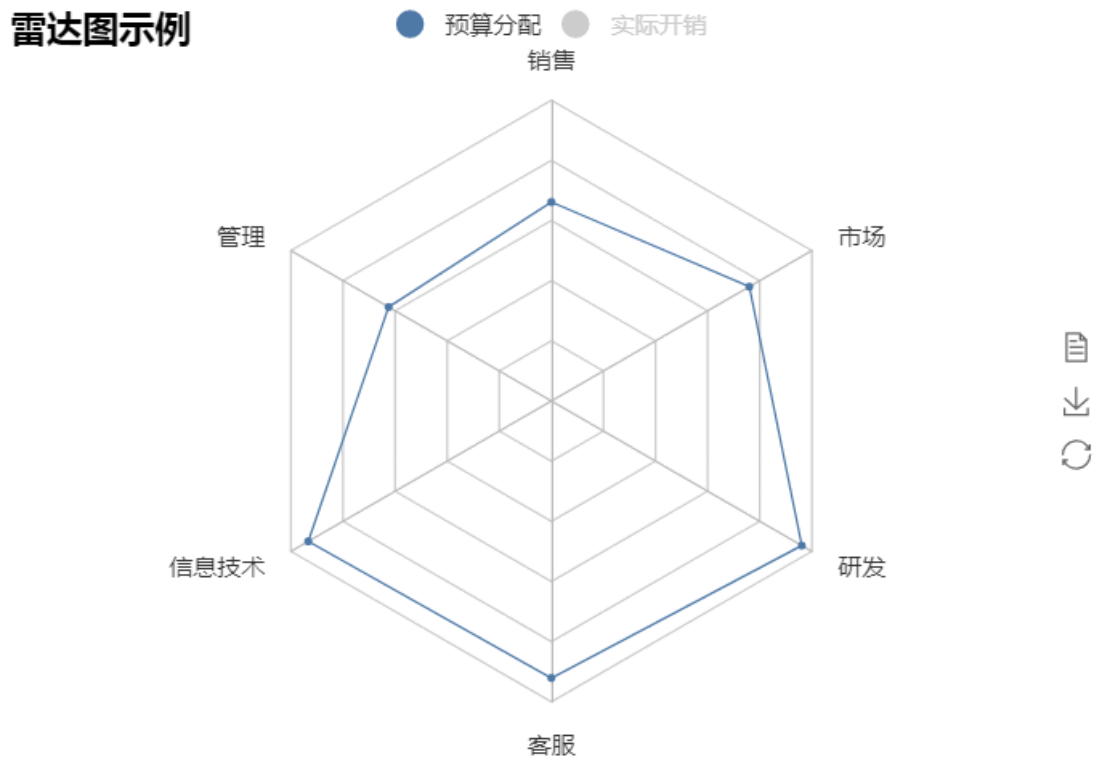
In [187]:

```
from pyecharts import Radar

schema = [
    ("销售", 6500), ("管理", 16000), ("信息技术", 30000), ("客服", 38000), ("研发", 52000), ("市场",
v1 = [[4300, 10000, 28000, 35000, 50000, 19000]]
v2 = [[5000, 14000, 28000, 31000, 42000, 21000]]
radar = Radar("雷达图示例", width=550)
radar.config(schema)
radar.add("预算分配", v1, is_splitline=True, is_axisline_show=True)
radar.add("实际开销", v2, label_color=["#4e79a7"], is_area_show=True, legend_selectedmode='single')

radar
```

Out[187]:



散点图示例

In [188]:

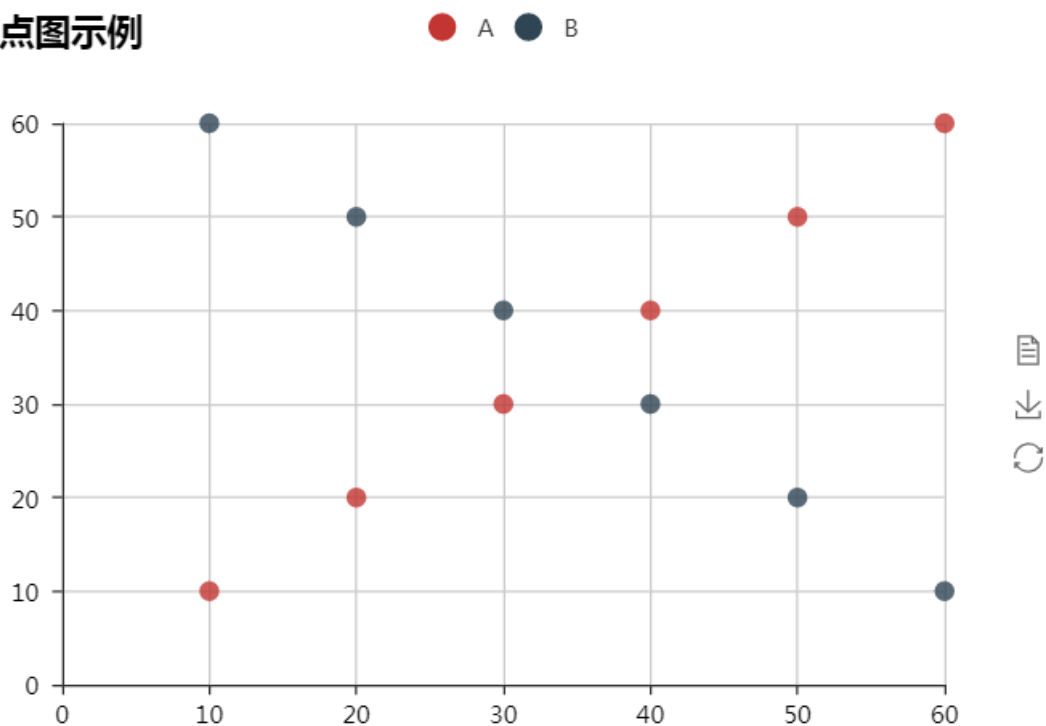
```
from pyecharts import Scatter

v1 = [10, 20, 30, 40, 50, 60]
v2 = [10, 20, 30, 40, 50, 60]
scatter = Scatter("散点图示例", width=550)
scatter.add("A", v1, v2)
scatter.add("B", v1[::-1], v2)

scatter
```

Out[188]:

散点图示例

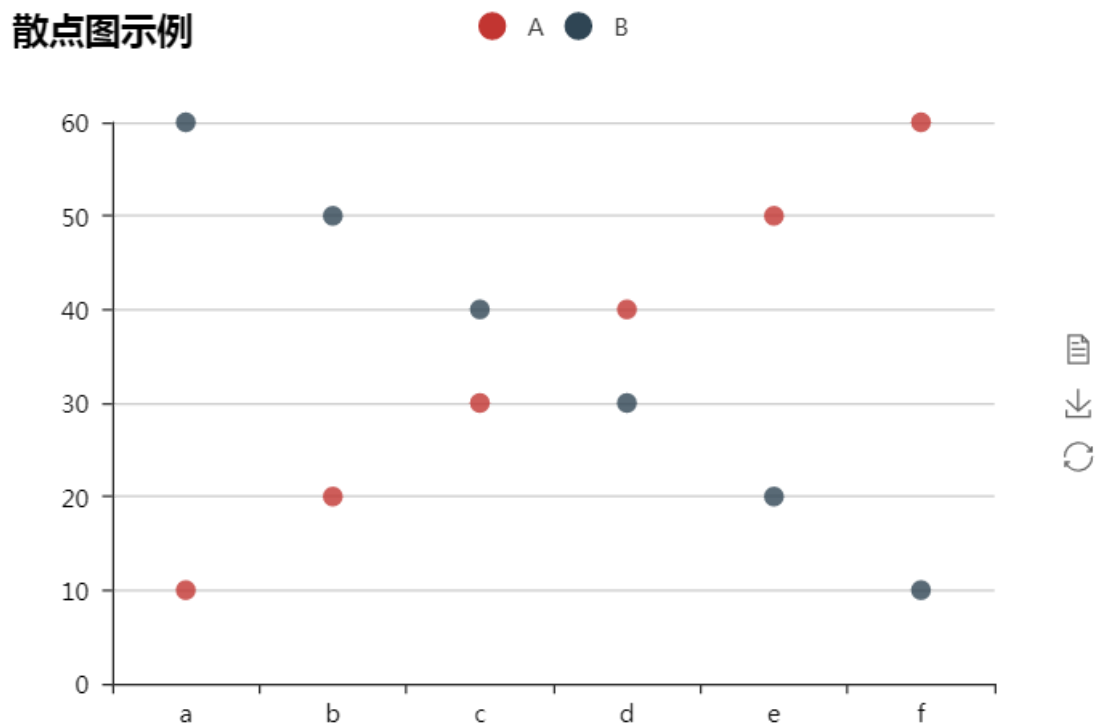


In [189]:

```
scatter = Scatter("散点图示例", width=550)
scatter.add("A", ["a", "b", "c", "d", "e", "f"], v2)
scatter.add("B", ["a", "b", "c", "d", "e", "f"], v1[::-1], xaxis_type="category")

scatter
```

Out[189]:



3D散点示例

In [190]:

```
from pyecharts import Scatter3D

import random

data = [[random.randint(0, 100), random.randint(0, 100), random.randint(0, 100)] for _ in range(80)]
range_color = ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf',
               '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026']
scatter3D = Scatter3D("3D 散点示例", width=550)
scatter3D.add("", data, is_visualmap=True, visual_range_color=range_color)

scatter3D
```

Out[190]:

3D 散点示例



词云

In [191]:

```
from pyecharts import WordCloud

name = ['Sam S Club', 'Macys', 'Amy Schumer', 'Jurassic World', 'Charter Communications',
        'Chick Fil A', 'Planet Fitness', 'Pitch Perfect', 'Express', 'Home', 'Johnny Depp',
        'Lena Dunham', 'Lewis Hamilton', 'KXAN', 'Mary Ellen Mark', 'Farrah Abraham',
        'Rita Ora', 'Serena Williams', 'NCAA baseball tournament', 'Point Break']
value = [10000, 6181, 4386, 4055, 2467, 2244, 1898, 1484, 1112, 965, 847, 582, 555,
         550, 462, 366, 360, 282, 273, 265]
wordcloud = WordCloud(width=550)
wordcloud.add("", name, value, word_size_range=[20, 100])

wordcloud
```

Out[191]:



多图

In [192]:

```

from pyecharts import Bar, Line, Scatter, EffectScatter, Grid

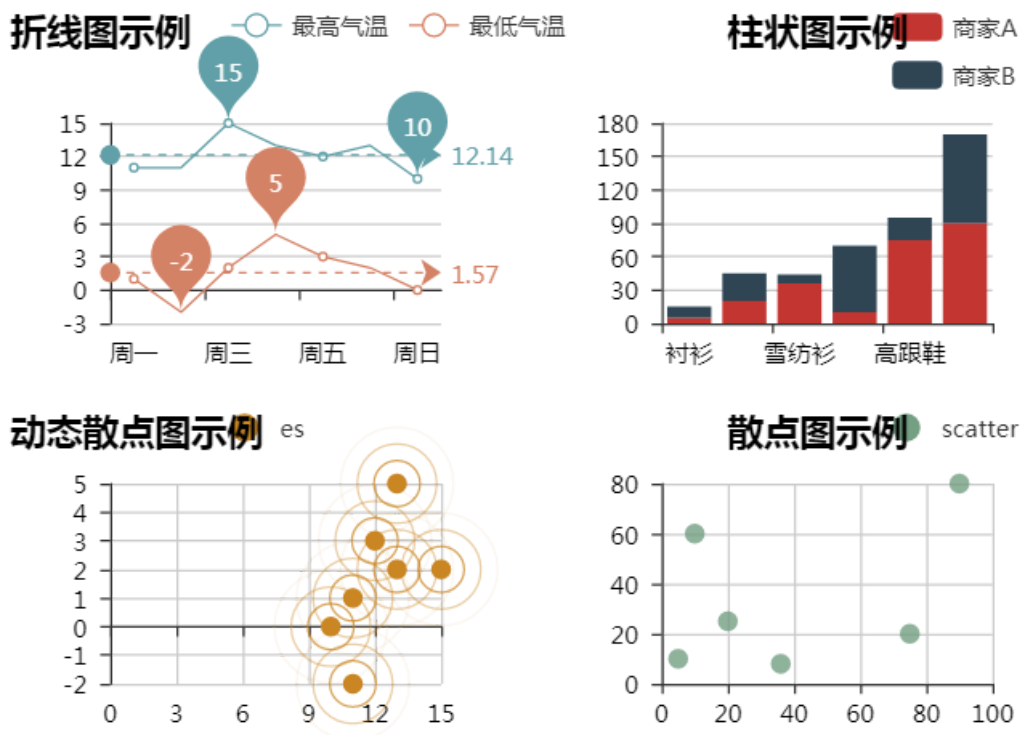
attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
v1 = [5, 20, 36, 10, 75, 90]
v2 = [10, 25, 8, 60, 20, 80]
bar = Bar("柱状图示例", height=720, width=1200, title_pos="65%")
bar.add("商家A", attr, v1, is_stack=True)
bar.add("商家B", attr, v2, is_stack=True, legend_pos="80%")
line = Line("折线图示例")
attr = ['周一', '周二', '周三', '周四', '周五', '周六', '周日']
line.add("最高气温", attr, [11, 11, 15, 13, 12, 13, 10], mark_point=["max", "min"], mark_line=["average"],
line.add("最低气温", attr, [1, -2, 2, 5, 3, 2, 0], mark_point=["max", "min"],
mark_line=["average"], legend_pos="20%")
v1 = [5, 20, 36, 10, 75, 90]
v2 = [10, 25, 8, 60, 20, 80]
scatter = Scatter("散点图示例", title_top="50%", title_pos="65%")
scatter.add("scatter", v1, v2, legend_top="50%", legend_pos="80%")
es = EffectScatter("动态散点图示例", title_top="50%")
es.add("es", [11, 11, 15, 13, 12, 13, 10], [1, -2, 2, 5, 3, 2, 0], effect_scale=6,
legend_top="50%", legend_pos="20%")

grid = Grid("多图", width=550)
grid.add(bar, grid_bottom="60%", grid_left="60%")
grid.add(line, grid_bottom="60%", grid_right="60%")
grid.add(scatter, grid_top="60%", grid_left="60%")
grid.add(es, grid_top="60%", grid_right="60%")

grid

```

Out[192]:



Line-Bar示例

In [194]:

```
from pyecharts import Bar, Line, Overlap

attr = ['A', 'B', 'C', 'D', 'E', 'F']
v1 = [10, 20, 30, 40, 50, 60]
v2 = [38, 28, 58, 48, 78, 68]

bar = Bar("Line-Bar 示例")
bar.add("bar", attr, v1)
line = Line()
line.add("line", attr, v2)

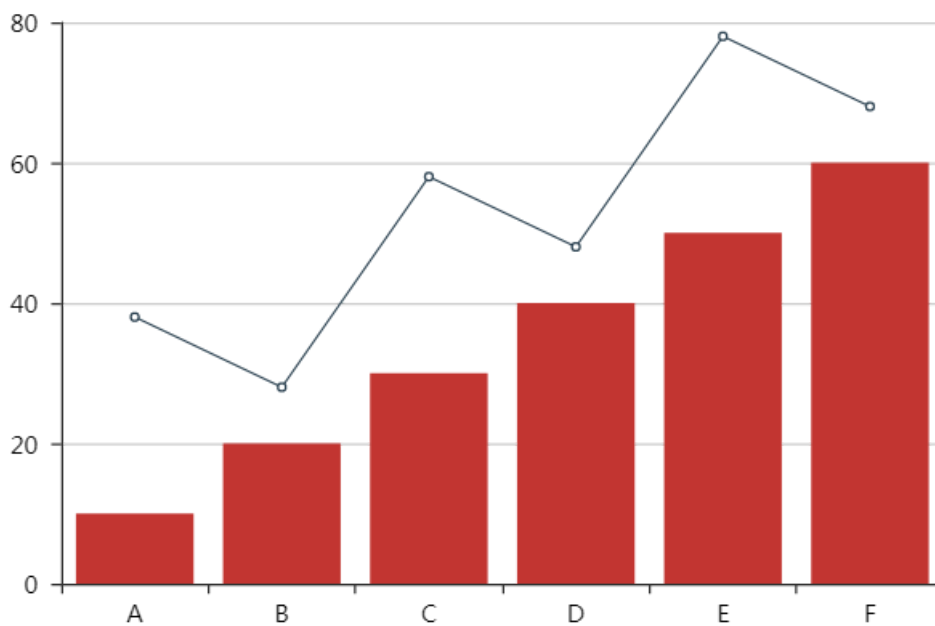
overlap = Overlap(width=550)
overlap.add(bar)
overlap.add(line)

overlap
```

Out[194]:

Line-Bar 示例

■ bar ○ line



In [195]:

```

from pyecharts import Bar, Timeline

from random import randint

attr = ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
bar_1 = Bar("2012 年销量", "数据纯属虚构")
bar_1.add("春季", attr, [randint(10, 100) for _ in range(6)])
bar_1.add("夏季", attr, [randint(10, 100) for _ in range(6)])
bar_1.add("秋季", attr, [randint(10, 100) for _ in range(6)])
bar_1.add("冬季", attr, [randint(10, 100) for _ in range(6)])

bar_2 = Bar("2013 年销量", "数据纯属虚构")
bar_2.add("春季", attr, [randint(10, 100) for _ in range(6)])
bar_2.add("夏季", attr, [randint(10, 100) for _ in range(6)])
bar_2.add("秋季", attr, [randint(10, 100) for _ in range(6)])
bar_2.add("冬季", attr, [randint(10, 100) for _ in range(6)])

bar_3 = Bar("2014 年销量", "数据纯属虚构")
bar_3.add("春季", attr, [randint(10, 100) for _ in range(6)])
bar_3.add("夏季", attr, [randint(10, 100) for _ in range(6)])
bar_3.add("秋季", attr, [randint(10, 100) for _ in range(6)])
bar_3.add("冬季", attr, [randint(10, 100) for _ in range(6)])

bar_4 = Bar("2015 年销量", "数据纯属虚构")
bar_4.add("春季", attr, [randint(10, 100) for _ in range(6)])
bar_4.add("夏季", attr, [randint(10, 100) for _ in range(6)])
bar_4.add("秋季", attr, [randint(10, 100) for _ in range(6)])
bar_4.add("冬季", attr, [randint(10, 100) for _ in range(6)])

bar_5 = Bar("2016 年销量", "数据纯属虚构")
bar_5.add("春季", attr, [randint(10, 100) for _ in range(6)])
bar_5.add("夏季", attr, [randint(10, 100) for _ in range(6)])
bar_5.add("秋季", attr, [randint(10, 100) for _ in range(6)])
bar_5.add("冬季", attr, [randint(10, 100) for _ in range(6)], is_legend_show=True)

timeline = Timeline(is_auto_play=True, timeline_bottom=0, width=550)
timeline.add(bar_1, "2012年")
timeline.add(bar_2, '2013 年')
timeline.add(bar_3, '2014 年')
timeline.add(bar_4, '2015 年')
timeline.add(bar_5, '2016 年')

timeline

```

Out[195]:

