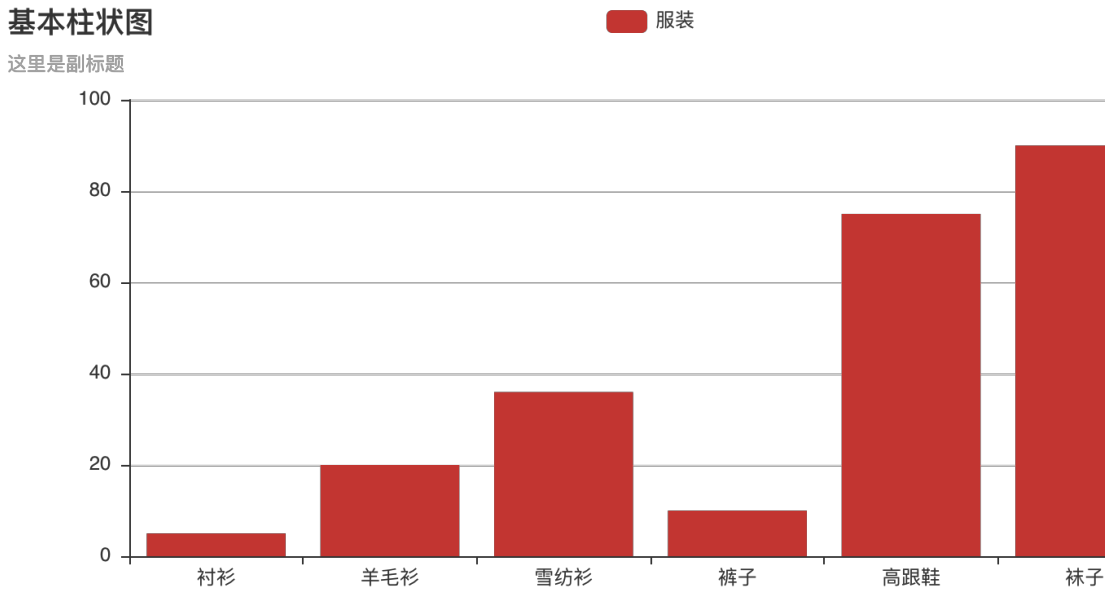
1.柱状图

In [28]:
```
from pyecharts import Bar
bar = Bar('基本柱状图', '这里是副标题')
bar.add('服装',['衬衫', '羊毛衫', '雪纺衫', '裤子', '高跟鞋', '袜子'],
        [5, 20, 36, 10, 75, 90],
        is_more_utils = True)
bar
```

Out[28]:



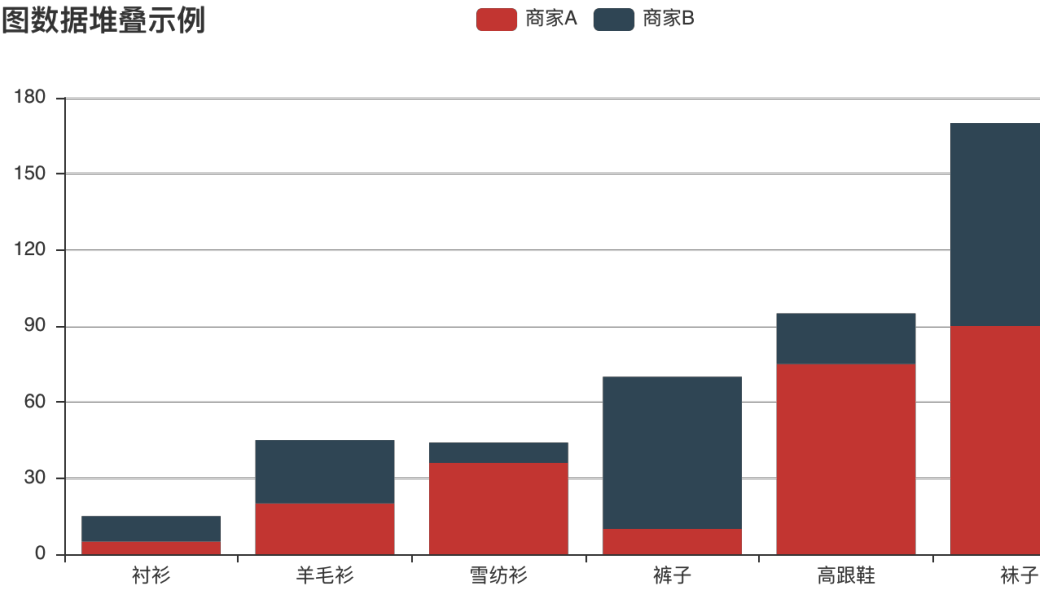
2.柱状堆叠图

is_stack是否堆叠

(1)is_stack = True

In [29]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 75, 90] # 商家A的各商品销量
v2 = [10, 25, 8, 60, 20, 80] # 商家B的各商品销量
# (1)添加标题
bar1 = Bar('柱状图数据堆叠示例')
# (2)添加数据
bar1.add('商家A', attr, v1, is_stack = True) # is_stack = False时, 就不堆叠了
bar1.add('商家B', attr, v2, is_stack = True)
# (3)展示堆叠图
bar1
```

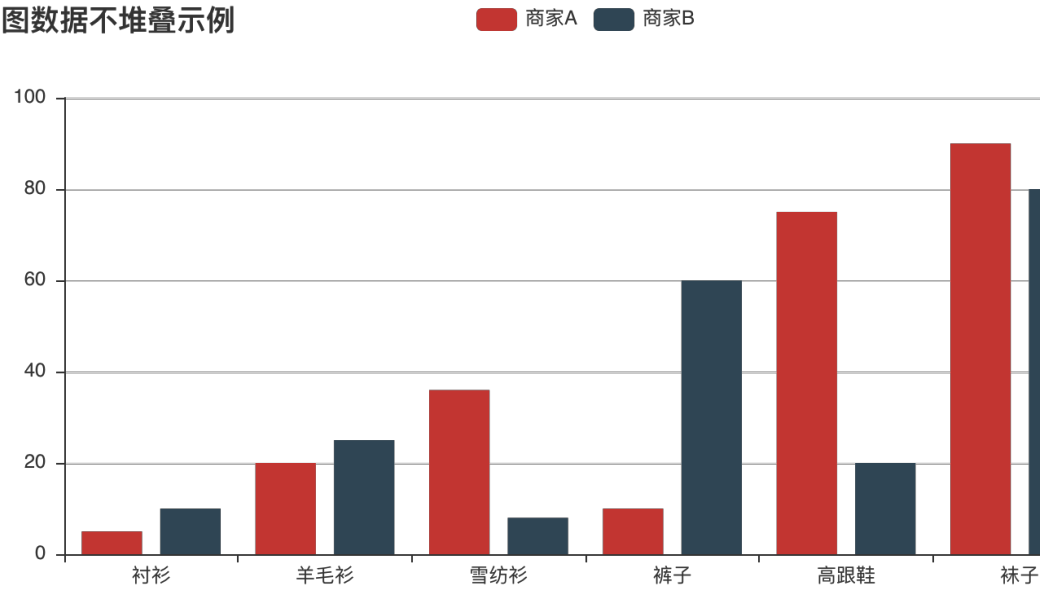
Out[29]:



(2)is_stack = None

In [30]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 75, 90] # 商家A的各商品销量
v2 = [10, 25, 8, 60, 20, 80] # 商家B的各商品销量
# (1)添加标题
bar1 = Bar('柱状图数据不堆叠示例')
# (2)添加数据
bar1.add('商家A', attr, v1, is_stack = False)
bar1.add('商家B', attr, v2, is_stack = False)
# (3)展示堆叠图
bar1
```

Out[30]:

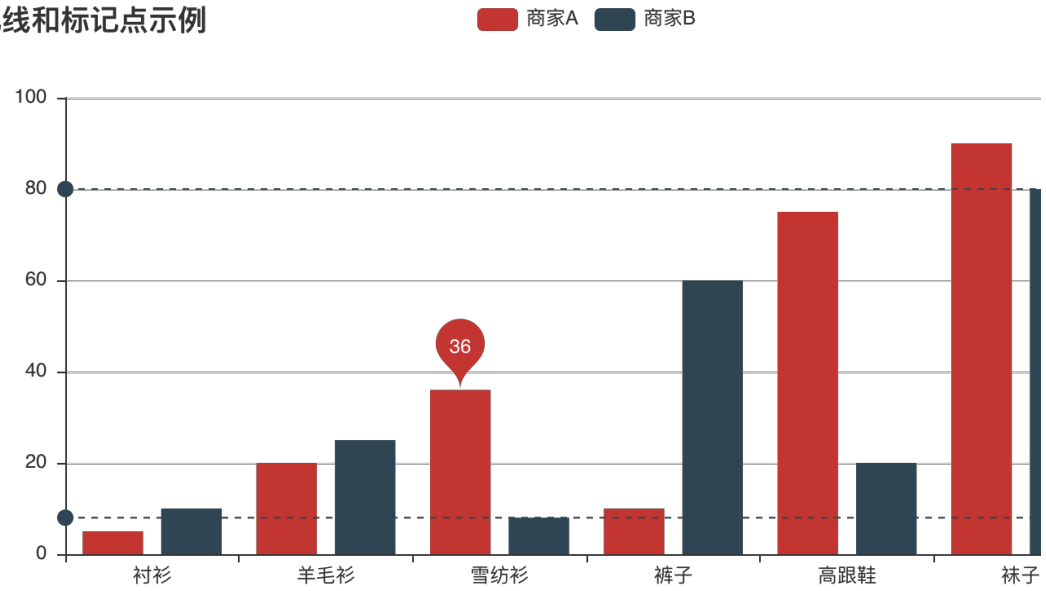


(3)标记线和标记点示例, mark_point, mark_line

In [31]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 75, 90] # 商家A的各商品销量
v2 = [10, 25, 8, 60, 20, 80] # 商家B的各商品销量
# (1)添加标题
bar = Bar('标记线和标记点示例')

# (2)添加数据
bar.add('商家A', attr, v1, mark_point = ['average']) # 标柱点
bar.add('商家B', attr, v2, mark_line = ['min', 'max']) # 标注线
bar
```

Out[31]:

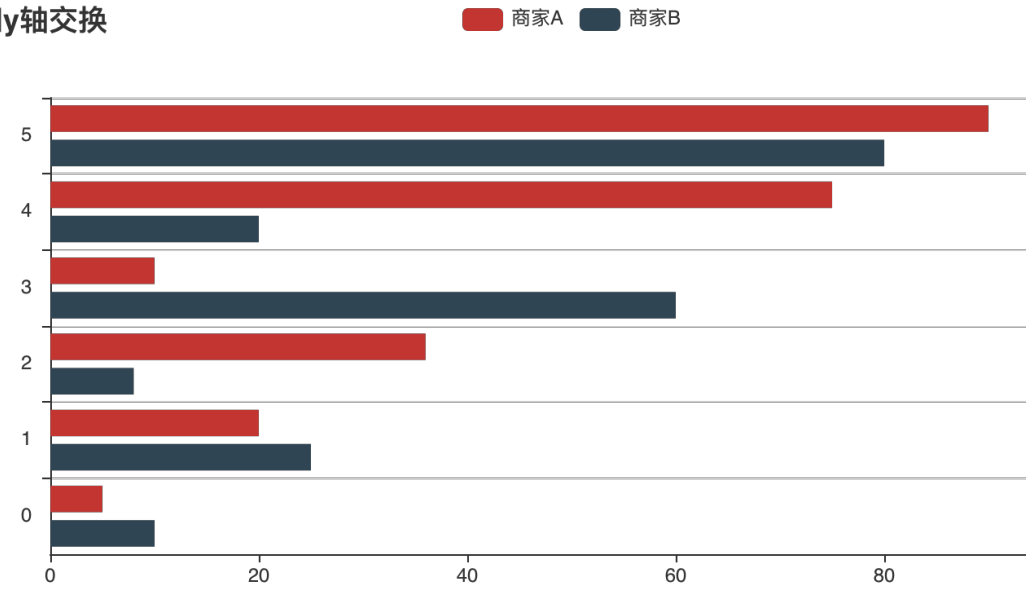


(4)x轴和y轴交换，is_convert,x轴的显示有问题

In [32]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 75, 90] # 商家A的各商品销量
v2 = [10, 25, 8, 60, 20, 80] # 商家B的各商品销量
# (1)添加标题
bar = Bar('x轴和y轴交换')

# (2)添加数据
bar.add('商家A', attr, v1)
bar.add('商家B', attr, v2, is_convert = True)
bar
```

Out[32]:



3.折线图

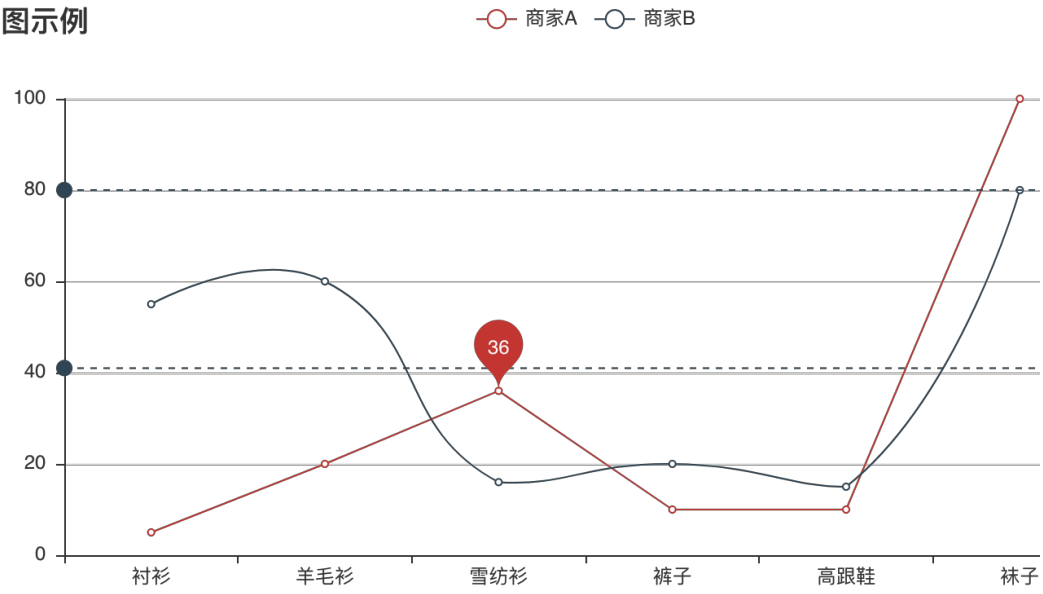
In [33]:
```
#(1)导入Line模块
from pyecharts import Line

attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 10, 100] # 商家A的各商品销量
v2 = [55, 60, 16, 20, 15, 80] # 商家B的各商品销量
# (1)添加标题
line = Line('折线图示例')

# (2)添加数据
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[33]:

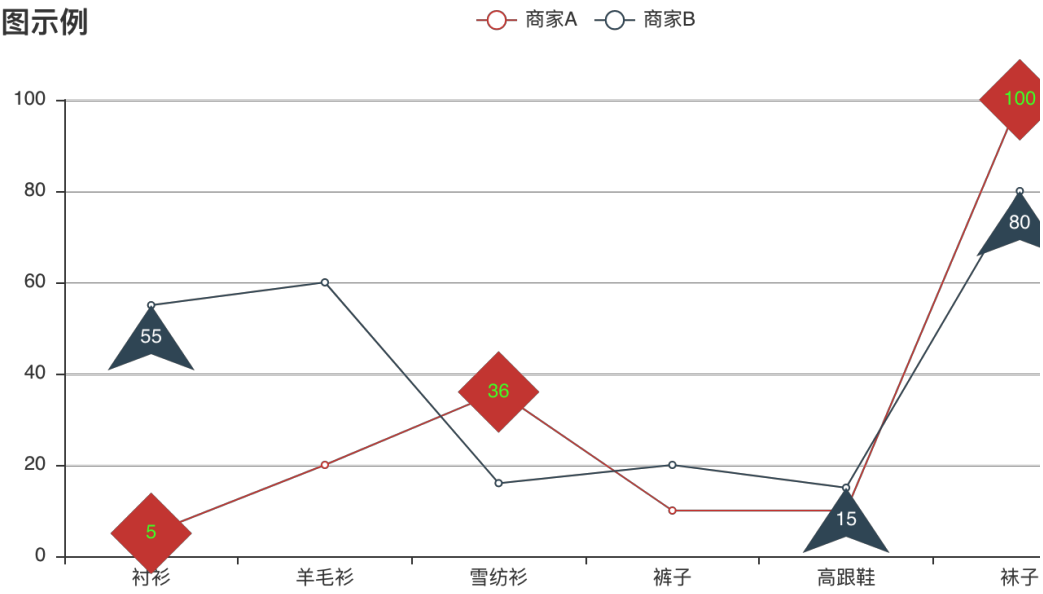


(1)折线图修改标注点的形状和标注文字的颜色大小

In [34]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 10, 100] # 商家A的各商品销量
v2 = [55, 60, 16, 20, 15, 80] # 商家B的各商品销量
# (1)添加标题
line = Line('折线图示例')

# (2)添加数据
line.add('商家A', attr, v1,
         mark_point = ['average', 'max', 'min'], # 添加标注点
        mark_point_symbol = 'diamond', # 设置标注点形状
        mark_point_textcolor = '#40ff27')  # 设置标注点颜色
line.add('商家B', attr, v2,
          mark_point=["average","max", "min"],
         mark_point_symbol='arrow',
         mark_point_symbolsize=40)
line
```

Out[34]:

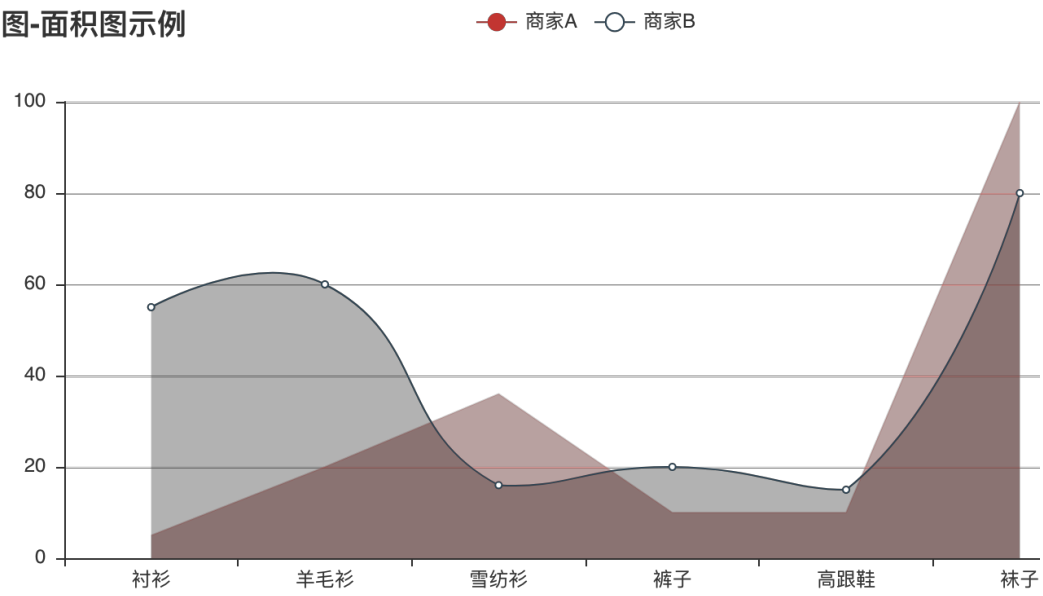


(2)折线图面积

In [35]:
```
attr = ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
v1 = [5, 20, 36, 10, 10, 100] # 商家A的各商品销量
v2 = [55, 60, 16, 20, 15, 80] # 商家B的各商品销量
# (1)添加标题
line = Line('折线图-面积图示例')

# (2)添加数据
line.add('商家A', attr, v1,
         is_fill = True, # 是否填充曲线所绘制面积
         line_opacity = 0.2, # 线条的不透明度
         area_opacity = 0.4, # 填充区域的不透明度
         symbol = None) #是否标注转折点
line.add('商家B', attr, v2,
         is_fill = True,
         area_color = '#000',
         area_opacity = 0.3,
         is_smooth = True)
line
```

Out[35]:

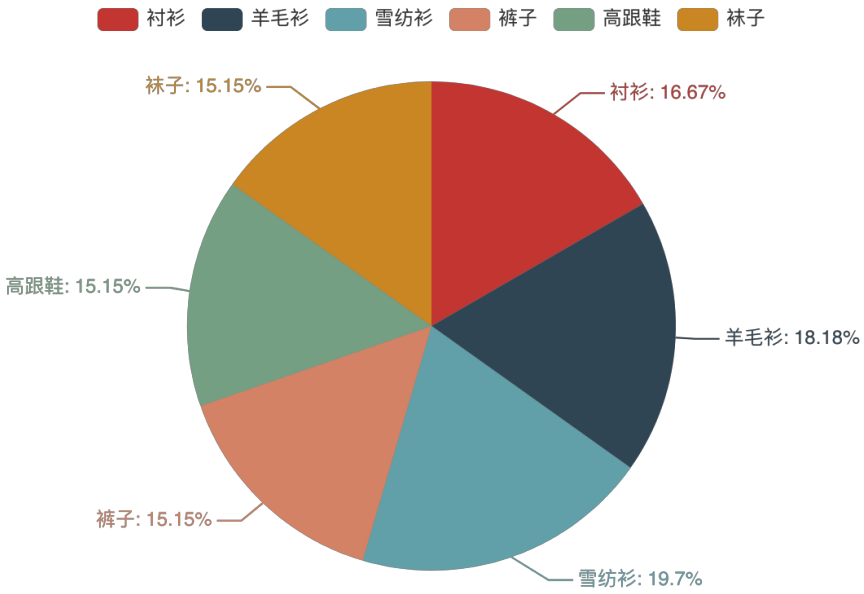


4.饼图

In [36]:
```
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')
pie.add('', attr, v1, is_label_show = True) # 是否显示标签
pie
```

Out[36]:

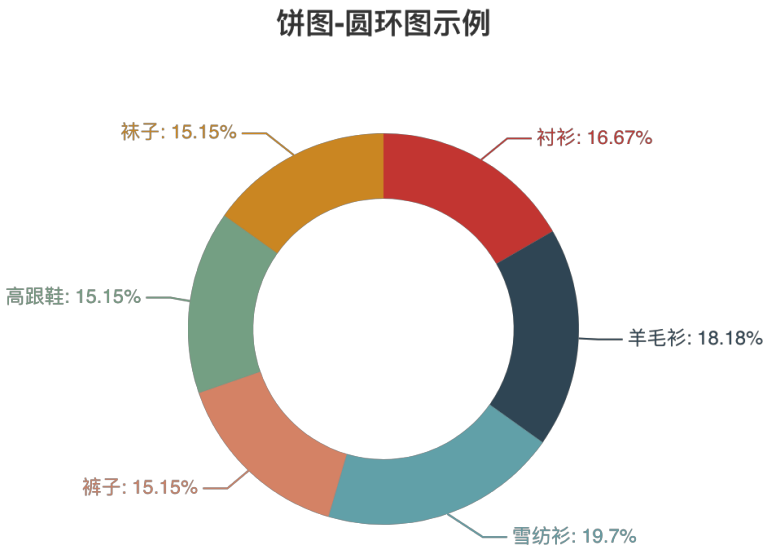


5.圆环图

In [37]:
```
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-圆环图示例', title_pos = 'center') # 标题位置设置
pie.add('haha', attr, v1, # "haha":鼠标放上去额外显示的标签
        radius = [40, 60], # 扇区圆心角展现数据的百分比，半径展现数据的大小,两个数应该分别是
        is_label_show = True,  # 是否显示标签
       legend_orient='vertical', #图例展开方向
       legend_pos='right') #图例的位置
pie
```

Out[37]:



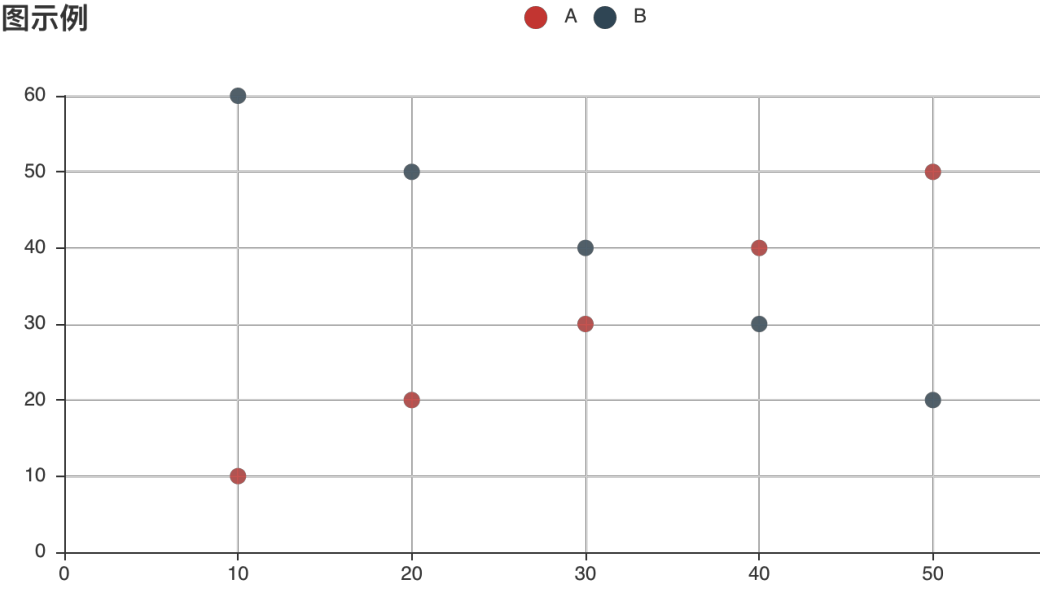
6.散点图

In [38]:
```
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')

scatter.add('A', v1, v2)
scatter.add('B', v1[::-1], v2)
scatter
```

Out[38]:

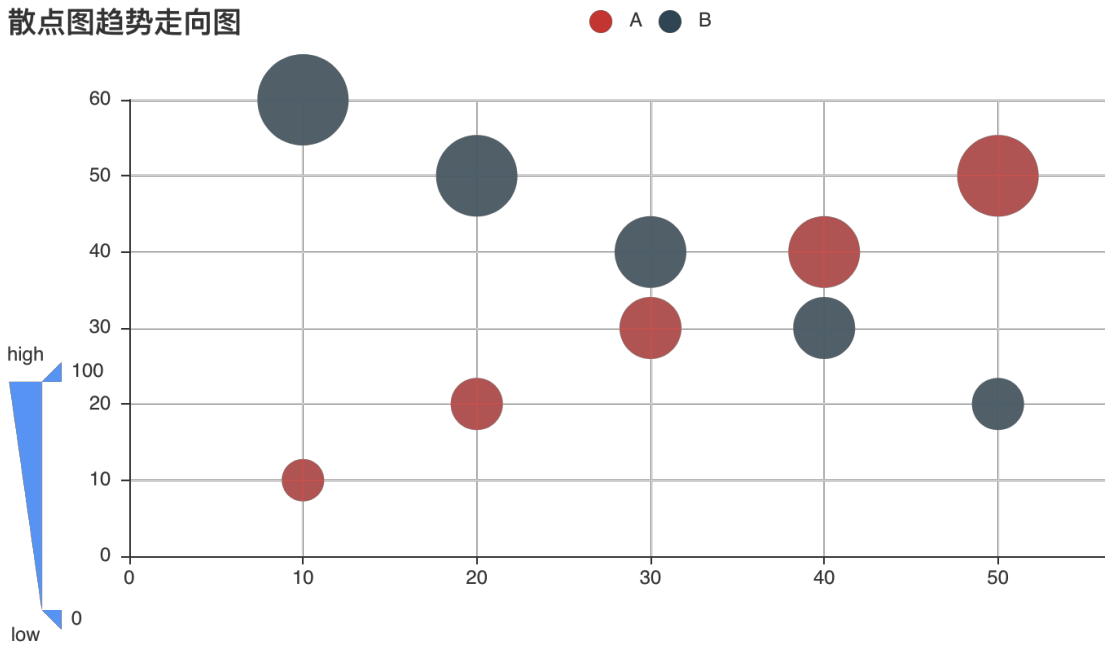


7.散点图趋势走向图

In [39]:
```
v1 = [10, 20, 30, 40, 50, 60]
v2 = [10, 20, 30, 40, 50, 60]
scatter = Scatter('散点图趋势走向图')

scatter.add('A', v1, v2)
scatter.add('B', v1[::-1], v2,
            is_visualmap = True, #是否展示趋势，is_visualmap=False时为散点图
            visual_type = 'size',
            visual_range_size = [20, 80])
scatter
```

Out[39]:



8.地图

(1)中国地图

In [40]:
```
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 = 1200, height = 600)
map.add('', attr, value, maptype = 'china', is_label_show = True)
map
```

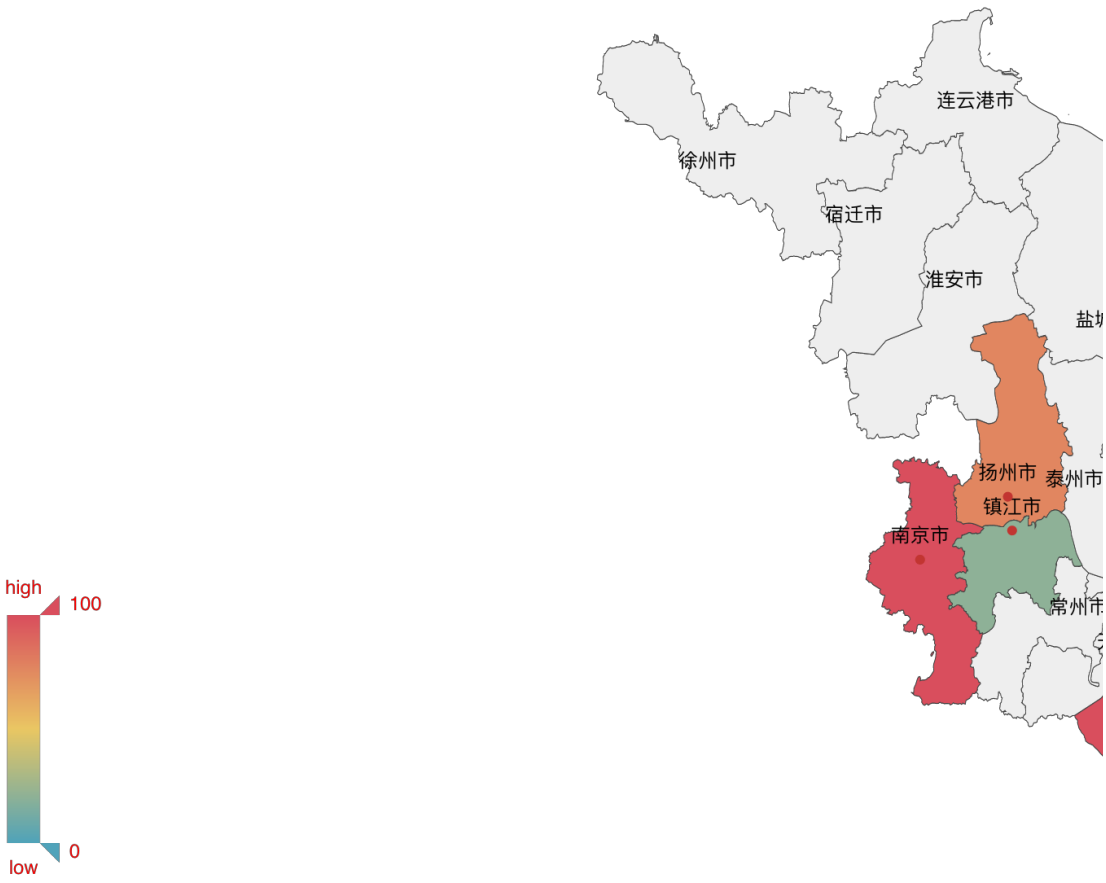
Out[40]:

全国地图示例

(2)江苏省地图

In [41]:
```
value = [20, 190, 253, 77, 65]
attr = ['镇江市', '南京市', '苏州市', '扬州市', '南通市']
map = Map('江苏省地图市例', width = 1200, height = 600)
map.add('', attr, value,
        maptype = '江苏',
        is_visualmap = True,
        visual_text_color = '#ff0000',
        is_label_show = True) # 是否显示额外的标签
map
```

Out[41]:

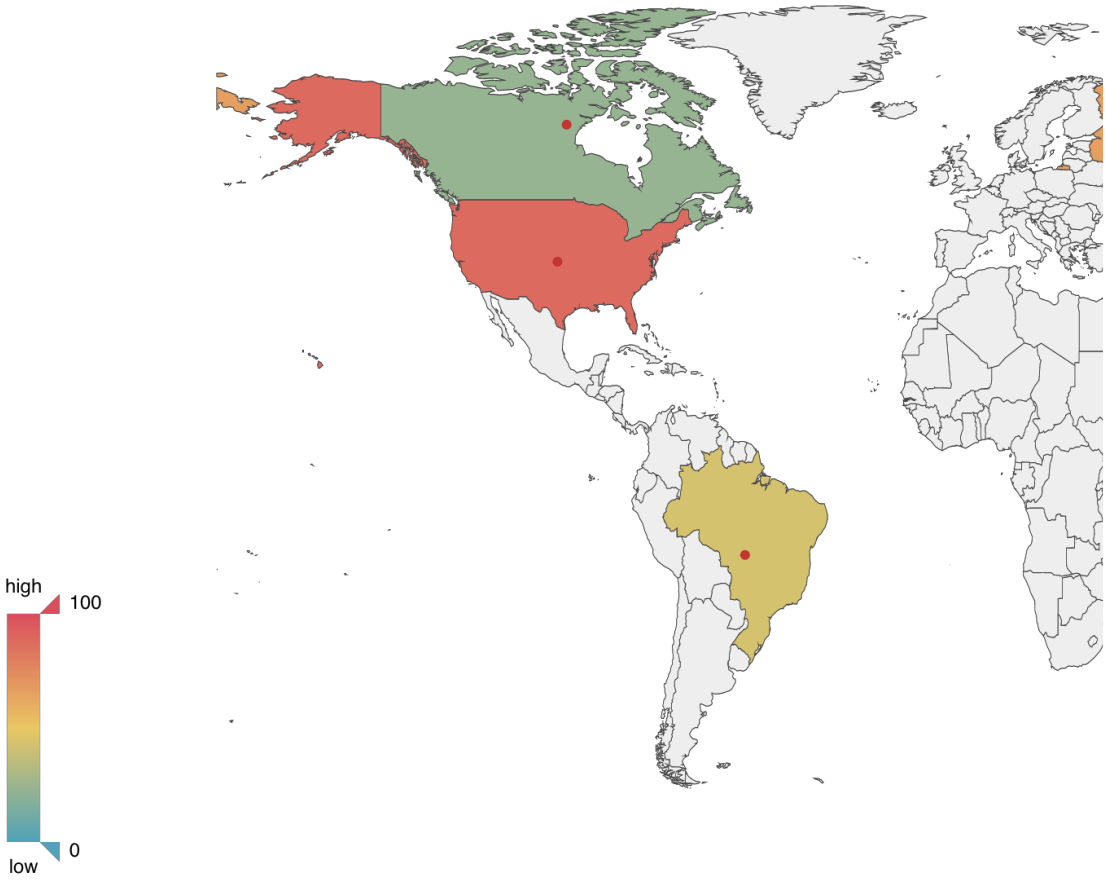


(3)世界地图

In [42]:
```
from pyecharts import Map
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 = 1200, height = 600)
map.add('', attr, value, maptype = 'world', is_visualmap = True,
        visual_text_color = '#000')
map
```

Out[42]:

世界地图示例



9.词云图

In [43]:
```
from pyecharts import WordCloud
name = ['Sam S Club', 'Macys', 'Amy Schumer', 'Jurassic World','Charter Communicat
        'Chick Fil A', 'Planet Fitness', 'Pitch Perfect', 'Express','Home','Johnny
        'Lena Dunham', 'Lewis Hamilton', 'KXAN', 'Mary EllenMark','Farrah Abraham'
        'Rita Ora', 'Serena Williams', 'NCAA baseball tournament','Point Break',
       '武汉','上海','广州','深圳','北京']
value = [10000, 6181, 4386, 4055, 2467,
         2244, 1898, 1484, 1112,965,
         847, 582, 555, 550, 462,
         366, 360, 282, 273, 265,
        120,150,180,185,195]
wordcloud = WordCloud(width = 1300, height = 620)
wordcloud.add('', name, value, word_size_range = [20, 100])
wordcloud
```

Out[43]:

10.水球图

(1)静态

In [44]:
```
from pyecharts import Liquid
liquid = Liquid('水球图示例')
liquid.add('Liquid', [0.6])
liquid
```

Out[44]: 水球图示例

(2)流动

In [45]:
```
from pyecharts import Liquid
liquid = Liquid('水球图示例-流动')
liquid.add('Liquid', [0.6, 0.5, 0.4, 0.3], is_liquid_outline_show = False)
liquid
```

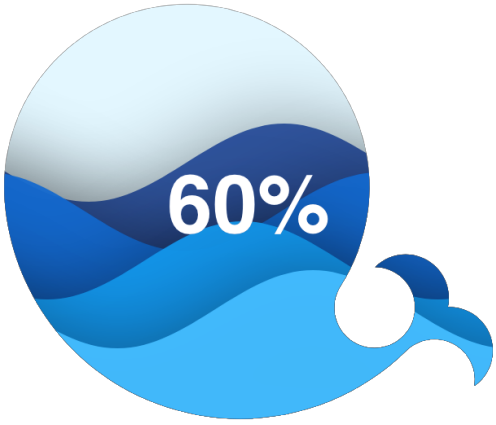
Out[45]: 水球图示例-流动

(3)流动， 鱼状

In [46]:
```
from pyecharts import Liquid
shape = ("path://M367.855,428.202c-3.674-1.385-7.452-1.966-11.146-1"
         ".794c0.659-2.922,0.844-5.85,0.58-8.719 c-0.937-10.407-7."
         "663-19.864-18.063-23.834c-10.697-4.043-22.298-1.168-29.9"
         "02,6.403c3.015,0.026,6.074,0.594,9.035,1.728 c13.626,5."
         "151,20.465,20.379,15.32,34.004c-1.905,5.02-5.177,9.115-9"
         ".22,12.05c-6.951,4.992-16.19,6.536-24.777,3.271 c-13.625"
         "-5.137-20.471-20.371-15.32-34.004c0.673-1.768,1.523-3.423"
         ",2.526-4.992h-0.014c0,0,0,0,0,0.014 c4.386-6.853,8.145-14"
         ".279,11.146-22.187c23.294-61.505-7.689-130.278-69.215-153"
         ".579c-61.532-23.293-130.279,7.69-153.579,69.202 c-6.371,"
         "16.785-8.679,34.097-7.426,50.901c0.026,0.554,0.079,1.121,"
         "0.132,1.688c4.973,57.107,41.767,109.148,98.945,130.793 c58."
         "162,22.008,121.303,6.529,162.839-34.465c7.103-6.893,17.826"
         "-9.444,27.679-5.719c11.858,4.491,18.565,16.6,16.719,28.643 "
         "c4.438-3.126,8.033-7.564,10.117-13.045C389.751,449.992,"
         "382.411,433.709,367.855,428.202z")
liquid = Liquid("水球图示例", width=1000, height=600)
liquid.add("Liquid", [0.6, 0.5, 0.4, 0.3],
           shape=shape, is_liquid_outline_show=False)
liquid
```

Out[46]:

水球图示例



四.overlap，图表叠加

1.步聚

将多张图表聚合到一个画板上，横坐标一样

add(chart,xaxis_index=0,
yaxis_index=0,
is_add_xaxis=False,
is_add_yaxis=False)
• chart -> chart instance：图表示例
• xaxis_index -> int：x 坐标轴索引，默认为 0
• yaxis_index -> int：y 坐标轴索引，默认为 0
• is_add_xaxis -> bool：是否新增一个 x 坐标轴，默认为 False

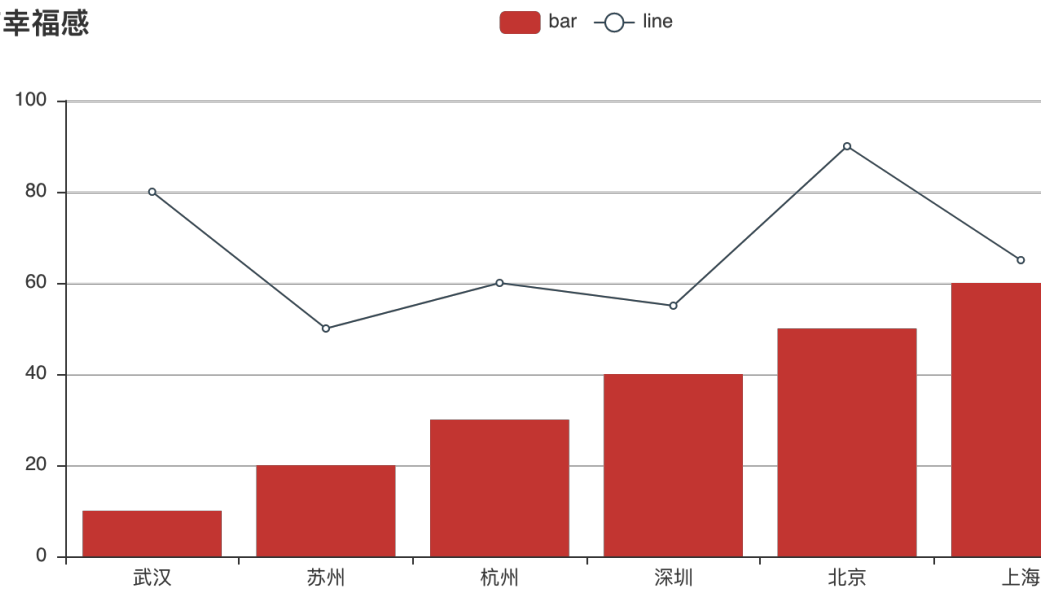
1.利用Overlap叠加Line + Bar

In [47]:
```
from pyecharts import Bar, Line, Overlap
attr = ['武汉', '苏州', '杭州', '深圳', '北京', '上海']
v1 = [10, 20, 30, 40, 50, 60]
v2 = [80, 50, 60, 55, 90, 65]

# 绘制条形图
bar = Bar('城市幸福感')
bar.add('bar', attr, v1)
# 绘制折线图
line = Line()
line.add('line', attr, v2)

# 聚合
overlap = Overlap()
overlap.add(bar)
overlap.add(line)
overlap
```

Out[47]:

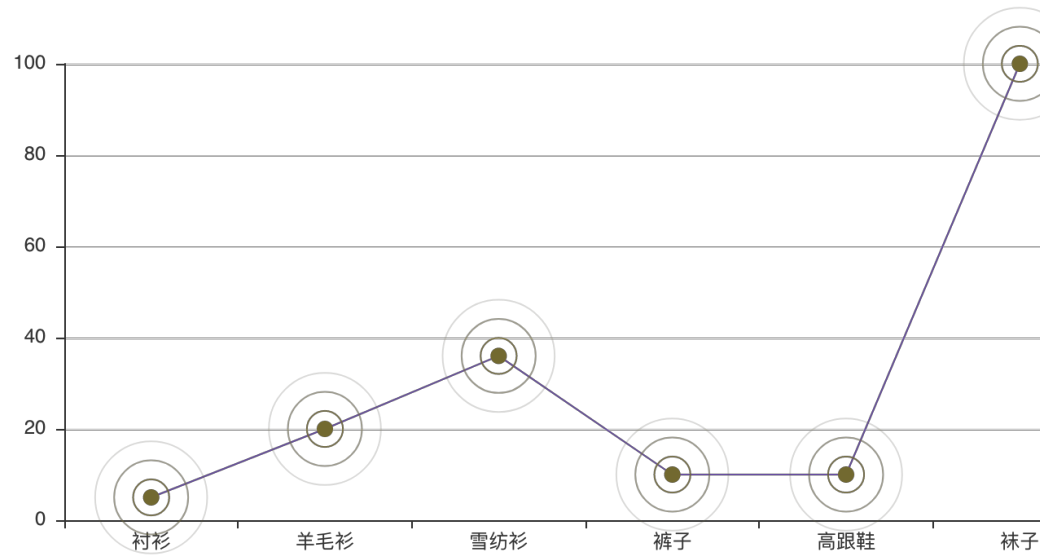


2.利用Overlap叠加Line + EffectScatter

In [49]:
```
from pyecharts import Line, EffectScatter, Overlap
attr = ['衬衫', '羊毛衫', '雪纺衫', '裤子', '高跟鞋', '袜子']
v1 = [5, 20, 36, 10, 10, 100]
line = Line('line-es示例')
line.add('', attr, v1, is_random = True)

es = EffectScatter('')
es.add('', attr, v1, effect_scale = 8)
overlap = Overlap()
overlap.add(line)
overlap.add(es)
overlap
```

Out[49]:



In [ ]: