



饼图组件

图表配置 (开发用)

基础配置:

ChartBase (基础配置):

```
{
  // 动画相关
  animation?: {
    durationX?: number | undefined;
    durationY?: number | undefined;
    easingX?: EasingType | undefined;
    easingY?: EasingType | undefined;
  } | undefined;

  chartBackgroundColor?: Color | undefined; // 图表背景色
  logEnabled?: boolean | undefined; // 是否启用chart日志输出, 性能损耗, 默认关闭
  noDataText?: string | undefined; // 没有数据时的提示文本

  touchEnabled?: boolean | undefined; // 是否开启触摸
  dragDecelerationEnabled?: boolean | undefined; // 是否启动滑动阻力
  dragDecelerationFrictionCoef?: number | undefined; // 滑动摩擦阻力系数

  highlightPerTapEnabled?: boolean | undefined; // 是否启用点击数据高亮
  // 图表描述属性
  chartDescription?: {
    text?: string | undefined; // 描述文本
    textColor?: Color | undefined; // 文本颜色
    textSize?: number | undefined; // 文本字体大小
    positionX?: number | undefined; // x轴位置
    positionY?: number | undefined; // y轴位置
  } | undefined;

  // 图例属性
  legend?: ChartLegend | undefined; // 见下方Legend

  // x轴属性, 与 yAxis 一样继承 Axis
  xAxis?: {
    ...Axis // 继承 Axis, 详见下部分
    labelRotationAngle?: number | undefined; // 标签旋转角度
    avoidFirstLastClipping?: boolean | undefined; // 是否避免首次和最后一次剪裁
    position?: 'TOP' | 'BOTTOM' | 'BOTH_SIDED' | 'TOP_INSIDE' | 'BOTTOM_INSIDE' | undefined; // x轴位置
    yOffset?: number | undefined; // 轴与标签间距
  } | undefined;

  // 弹出标记属性
  marker?: {
    enabled?: boolean | undefined; // 启用
    digits?: number | undefined; // 数字
    markerColor?: Color | undefined; // 标记颜色
    textColor?: Color | undefined; // 字体颜色
    textSize?: number | undefined; // 字体大小
  } | undefined;

  // 高亮显示区域
  highlights?: Array<{
    x: number; // 饼状图实体下标
    dataSetIndex?: number | undefined; // 柱状图下标
    dataIndex?: number | undefined; // 组合图形时的下标
    y?: number | undefined;
    stackIndex?: number | undefined;
  }> | undefined;

  onSelect?: ((event: ChartSelectEvent) => void) | undefined;
  onChange?: ((event: ChartChangeEvent) => void) | undefined;
}
```

Legend (图例):

```
{
  enabled?: boolean | undefined; // 是否启用
  textColor?: Color | undefined; // 文字颜色
  textSize?: number | undefined; // 文字大小
  fontFamily?: string | undefined; // 字体
  fontStyle?: number | undefined; // 字体样式
}
```

```

fontWeight?: number | undefined; // 字体权重

wordWrapEnabled?: boolean | undefined; // 是否自动换行
maxSizePercent?: number | undefined; // 图例与图表的间距百分比

horizontalAlignment?: 'LEFT' | 'CENTER' | 'RIGHT' | undefined; // 水平对齐方式
verticalAlignment?: 'TOP' | 'CENTER' | 'BOTTOM' | undefined; // 垂直对齐方式
orientation?: 'HORIZONTAL' | 'VERTICAL' | undefined; // 图例和图表分布方向
drawInside?: boolean | undefined; // 图例是否绘制在图表内
direction?: 'LEFT_TO_RIGHT' | 'RIGHT_TO_LEFT' | undefined; // 图例的文字和图形分布方向

form?: 'NONE' | 'EMPTY' | 'DEFAULT' | 'SQUARE' | 'CIRCLE' | 'LINE' | undefined; // 图例形状
formSize?: number | undefined; // 图例大小
xEntrySpace?: number | undefined; // 图例间水平距离
yEntrySpace?: number | undefined; // 图例间垂直距离
formToTextSpace?: number | undefined; // 图例图形与文字间距

// 自定义图例颜色和标签
custom?: {
  colors?: Color[] | undefined;
  labels?: string[] | undefined;
} | undefined;
}

```

Axis (轴父类) :

```

{
  enabled?: boolean | undefined; // 启用
  drawLabels?: boolean | undefined; // 绘制轴标签 (刻度)
  drawAxisLines?: boolean | undefined; // 是否绘制轴线 (x轴或y轴的一条线)
  drawGridLines?: boolean | undefined; // 是否绘制网格线

  textColor?: Color | undefined; // 文字颜色
  textSize?: number | undefined; // 文字大小
  fontFamily?: string | undefined; // 文字字体
  fontStyle?: string | undefined; // 文字样式
  fontWeight?: number | undefined; // 文字权重
  gridColor?: Color | undefined; // 网格线颜色
  gridLineWidth?: number | undefined; // 网格线宽度
  axisLineColor?: Color | undefined; // 轴线颜色
  axisLineWidth?: number | undefined; // 轴线宽度
  // 网格虚线
  gridDashedLine?: {
    lineLength?: number | undefined; // 线长
    spaceLength?: number | undefined; // 间距
    phase?: number | undefined; // 阶段
  } | undefined;
  // 限制线, 一般用来表示平均值等
  limitLines?: {
    limit: number; // 线的坐标
    label?: string | undefined; // 标签文字
    lineColor?: Color | undefined; // 线颜色
    lineWidth?: number | undefined; // 线宽度
    valueTextColor?: Color | undefined; // 文本内容颜色
    valueFont?: string | undefined; // 字体值
    labelPosition?: 'LEFT_TOP' | 'LEFT_BOTTOM' | 'RIGHT_TOP' | 'RIGHT_BOTTOM' | undefined;
  } | undefined;
  // 标签位置
  lineDashPhase?: number | undefined; // 虚线偏移
  lineDashLengths?: number[] | undefined; // 虚线长度
} | undefined;
drawLimitLinesBehindData?: boolean | undefined; // 将限制线绘制在数据之下, 默认false

/**
 * 轴最大值和最小值决定了轴的开始和结束刻度坐标 (与visibleRange可视范围不同的是, 可视范围决定默认
 * 图表内可观察到的数据:
 * 1) 如果可视范围大于轴最大值, 则图表只能显示轴最大值范围内的数据, 超过轴最大值的数据需要拖动图表
 * 才可见);
 * 2) 如果轴最大值超过可视范围, 则只能在可视范围内选择数据, 超过可视范围无法操作;
 */
axisMaximum?: number | undefined; // 轴最小值
axisMinimum?: number | undefined; // 轴最大值

/**
 * granularityEnabled: 启用/禁用轴值间隔的粒度控制。如果启用, 则不允许轴间隔低于某个粒度。
 * 默认值:false。
 *
 * granularity: 轴间隔粒度, 在缩放时为轴设置一个最小间隔。轴间距不能低于这个极限。这可以用来避免
 * 标签缩放时重复显示。
 *
 * 注意! 注意! 注意!: 仅当强制指定确切标签数为false (labelCountForce=false) 时候生效。
 * 如果坐标轴是date时间为单位, 需要根据timeUnit指定单位来设置granularity间隔值:
 * 1) timeUnit单位是毫秒
 * 轴想精确到1秒后不允许继续放大, 则granularity可设置间隔为1000(毫秒);
 * 轴想精确到1分钟后不允许继续放大, 则granularity可设置为60*1000)
 * 2) timeUnit单位是秒
 * 轴想精确到1秒后不允许继续放大, 则granularity可设置间隔为1;
 * 轴想精确到1分钟后不允许继续放大, 则granularity可设置为60)
 */
granularity?: number | undefined;
granularityEnabled?: boolean | undefined;

/**
 * labelCountForce : 是否强制使用指定的标签数
 * true-将强制设置标签计数, 这意味着确切指定的标签计数将被强制执行;

```

```
*fase-绘制并沿着轴均匀分布—这可能导致标签具有不均匀的值;
*/
labelCount?: number | undefined;
labelCountForce?: boolean | undefined;

/**
*将轴标签居中，而不是将它们画在原来的位置。这对于分组的条形图尤其有用
(居于条形图的底部剧中显示，而不是靠近分割线之间显示刻度标签)
*/
centerAxisLabels?: boolean | undefined;
// 数据格式
valueFormatter?: ('largeValue' | 'percent' | 'date') | string | string[]; | undefined;

// 当valueFormatter格式是日期时候，需要指定valueFormatterPattern规则以及since, timeUnit
valueFormatterPattern?: string | undefined;
since?: number | undefined; // 当前开始时间: milliseconds from 1970-1-1 when x=0
timeUnit?: 'MILLISECONDS' | 'SECONDS' | 'MINUTES' | 'HOURS' | 'DAYS' | undefined;
// timeUnit of x, 时间戳
}
```

Dataset（数据集合）：

```
{
  label?: string | undefined; // 实体文本
}
```

CommonDataSetConfig（数据集合通用config）：

```
//继承属性ChartDataSetConfig.chartDataSetConfig
common: {
  color: PropTypes.number, //标注的图形的颜色
  colors: PropTypes.arrayOf(PropTypes.number), //标注的图形颜色（多个图形）
  highlightEnabled:PropTypes.bool, //数据点是否可点击
  drawValues: PropTypes.bool, //是否显示数据点上数字（y值）
  valueTextSize:PropTypes.number, //数据点上面数字大小
  valueTextColor:PropTypes.number, //数据点上面数字颜色
  visible:PropTypes.bool, //是否显示数据点图形
  //数据点图形上面的数字格式（大数 百分比 日期）
  valueFormatter: PropTypes.oneOfType([
    PropTypes.oneOf(['largeValue', 'percent', 'date']),
    PropTypes.string,
    PropTypes.arrayOf(PropTypes.string)
  ]),
  valueFormatterPattern: PropTypes.string, //数据格式正则
  axisDependency:PropTypes.oneOf(['LEFT', 'RIGHT']) //轴的依赖（左侧/右侧）
},
```

图表配置：

折线 / 柱状图：

```
{
  data: {
    datasets?: { // 数据集
      values?: { // 数据值
        x?: number | undefined; // x坐标
        y?: number; // y坐标
        marker?: string | undefined; // 弹出的标记属性
      }[] | number[] | undefined;
    }
    label?: string | undefined; // 图例的标签属性
    config?: { // 组件data属性内的config属性
      // 一些继承属性
      ...CommonDatasetConfig;
      ...BarLineScatterCandleBubbleConfig;
      ...LineScatterCandleRadarConfig;
      ...LineRadarConfig;
      circleRadius?: number | undefined; // 是否绘制折线上的数据点(圆)
      drawCircles?: boolean | undefined; // 是否绘制折线上数据点（圆）的半径（drawCircle=true有效）
      mode?: 'LINEAR' | 'STEPPED' | 'CUBIC_BEZIER' | 'HORIZONTAL_BEZIER' | undefined; // 折线样式
      drawCubicIntensity?: number | undefined; // 折线弯曲度（0-1），仅当mode为贝塞尔模式时生效
      circleColor?: Color | undefined; // 折线上数据点的颜色
      circleColors?: Color[] | undefined; // 折线上多个数据点的颜色
      circleHoleColor?: Color | undefined; // 折线数据点中心圆孔颜色(drawCircleHole=true有效)
      drawCircleHole?: boolean | undefined; // 是否绘制线上数据圆中心圆孔
      dashedLine?: { // 虚线样式
```

```
        lineLength: number; // 线长
        spaceLength: number; // 间距
        phase?: number | undefined;
      } | undefined;
    }
  }[]
}
```

饼图:

```
{
  // 继承PieRadarChartBase, 见下方PieRadarChartBase
  extends PieRadarChartBase

  // 见下方PieData
  data?: PieData | undefined;

  drawEntryLabels?: boolean | undefined; // 是否显示实体数据中的label, 搭配data-->config-->xValuePosition、ValuePosition:"OUTSIDE_SLICE"使用
  usePercentValues?: boolean | undefined; // 使用百分比数据格式, 搭配data-->config-->valueFormatter使用

  centerText?: string | undefined; // 环图中心的标签文本
  styledCenterText?: { // 有样式的环图中心标签文本
    text?: string | undefined;
    color?: Color | undefined;
    size?: number | undefined;
  } | undefined;
  centerTextRadiusPercent?: number | undefined; // 中心标签文本半径百分比 (决定排版方向和换行?)
  holeRadius?: number | undefined; // 饼图中心孔半径 (搭配transparentCircleRadius(要大于该值)可实现交界处阴影效果)
  holeColor?: Color | undefined; // 饼图中心孔背景颜色
  transparentCircleRadius?: number | undefined; // 透明圆半径
  transparentCircleColor?: Color | undefined; // 透明圆颜色

  entryLabelColor?: Color | undefined; // 实体值文本颜色
  entryLabelTextSize?: number | undefined; // 实体值文本大小
  maxAngle?: number | undefined; // 最大角度 (决定走后一个实体距离第一个实体的间距空隙)
}
```

PieRadarChartBase:

```
{
  // 继承ChartBase, 见最上方
  extends ChartBase

  minOffset?: number | undefined;
  rotationEnabled?: boolean | undefined; // 启用旋转
  rotationAngle?: number | undefined; // 旋转角度 (第一个实体默认旋转多少角度来展示)
}
```

PieData:

```
{
  dataSets?: {
    values?: {
      value: number; // 实体值 (必填)
      label?: string | undefined; // 实体文字标签
    }[] | number[];
    label: string // 饼图整体标签
    config?: {
      sliceSpace?: number | undefined; // 实体之间的间距
      selectionShift?: number | undefined; // 选中片段凸显高度 (值越大图表越小, 凸显越明显)
      xValuePosition?: PieValuePosition | undefined; // 实体文字标签显示位置
      yValuePosition?: PieValuePosition | undefined; // 实体数值标签显示位置
      /**
       * 实体标签指向实体区域的内线和外线长度 (仅当xValuePosition或yValuePosition有一个
       * 为OUTSIDE_SLICE时有效)。
       * Part1: 内线
       * Part2: 外线
       */
      valueLinePart1Length?: number | undefined;
      valueLinePart2Length?: number | undefined;
      valueLineColor?: Color | undefined; // 实体标签指向实体区域线的颜色
      valueLineWidth?: number | undefined; // 实体标签指向实体区域线的宽度
      /**
       * 实体标签指向实体区域内线(Part1)偏移实体片段百分比默认75.f
       * 0-内线以实体最内层区域作为起点;
       * 50-内线以实体中心区域作为起点;
       * 100-内线以实体最外层区域作为起点)
       */
      valueLinePart1OffsetPercentage?: number | undefined;
      valueLineVariableLength?: boolean | undefined; // 实体标签外线长度可变 (自适用)
    } extends CommonDataSetConfig // 见上方公共配置CommonDataSetConfig
  } extends Dataset [] | undefined;
}
```

图表能力总结

折线图：

雷达图：

环图：

饼图组件

属性名	值	描述
width	number	容器宽度
height	height	容器高度
showLegend	boolean	是否展示图例
centerText	string	饼图中心的标签文本
data	{ value: number, label: string }[]	图表使用的数据，value为实体值，label为实体标签
showLabel	boolean 'INSIDE' 'OUTSIDE'	是否展示data中的label，若为true则默认展示在外侧，也可通过字符串控制展示位置，若data为number[]则无效
showValue	boolean 'INSIDE' 'OUTSIDE'	是否展示data中的value，若为true则默认展示在外侧，也可通过字符串控制展示位置，若data为number[]则无效
percentage	boolean	使用百分比格式（需要数据支持）
rotationEnabled	boolean	是否开启旋转
usePie	boolean	是否使用饼图模式，默认false展示为环图
labelColor	ReturnType<typeof processColor>	实体标签文本的颜色
themeColors	ReturnType<typeof processColor>[]	注册的主题颜色，组件内部会按value由大到小排序后依次选取
onSelect	(entry: number { label: string, value: number }) => void	若 data 为 number[]，entry 为 value: number 若 data 为 { label: string, value: number }[]，entry为 { label, value }

```
interface IPieGraphProps {
  width?: number;
  height: number;
  showLegend?: boolean;
  centerText?: string;
  data: {
    label: string;
    value: number
  }[] | number[];
  showLabel?: boolean | EValuePosition;
  showValue?: boolean | EValuePosition;
  percentage?: boolean | undefined;
  rotationEnabled?: boolean;
  usePie?: boolean;
  labelColor?: ReturnType<typeof processColor>;
  themeColors?: ReturnType<typeof processColor>[];
  onSelect?: (entry: number | { label: string, value: number }) => void;
};
```

示例数据：

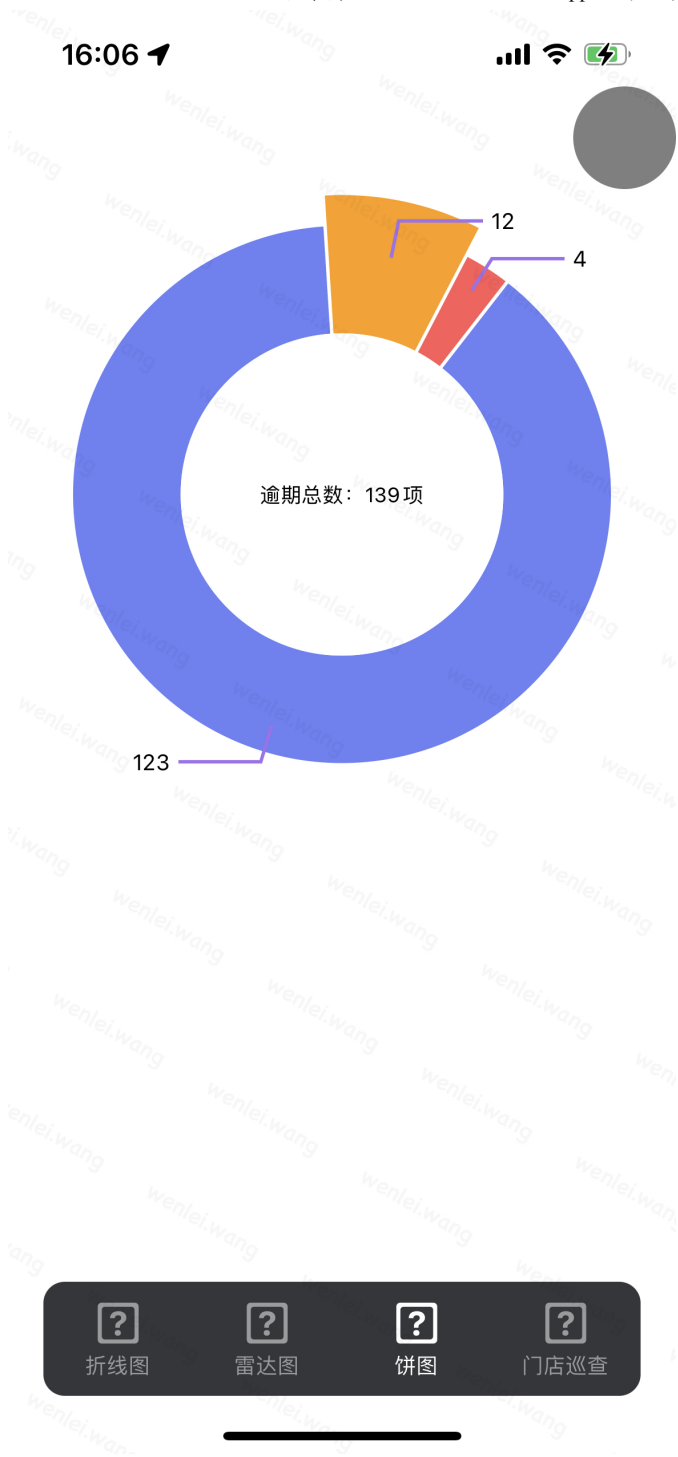
```
const testData = [
  {
    value: 12,
    label: 'P1任务',
  },
  {
    value: 4,
    label: 'P2任务',
  },
  {
    value: 123,
    label: 'P3任务',
  },
];

const simpleData = [46, 21, 89];

const percentData = [
  {
    value: 0.82,
    label: 'P1任务',
  },
  {
    value: 0.1,
    label: 'P2任务',
  },
  {
    value: 0.08,
    label: 'P3任务',
  },
];
```

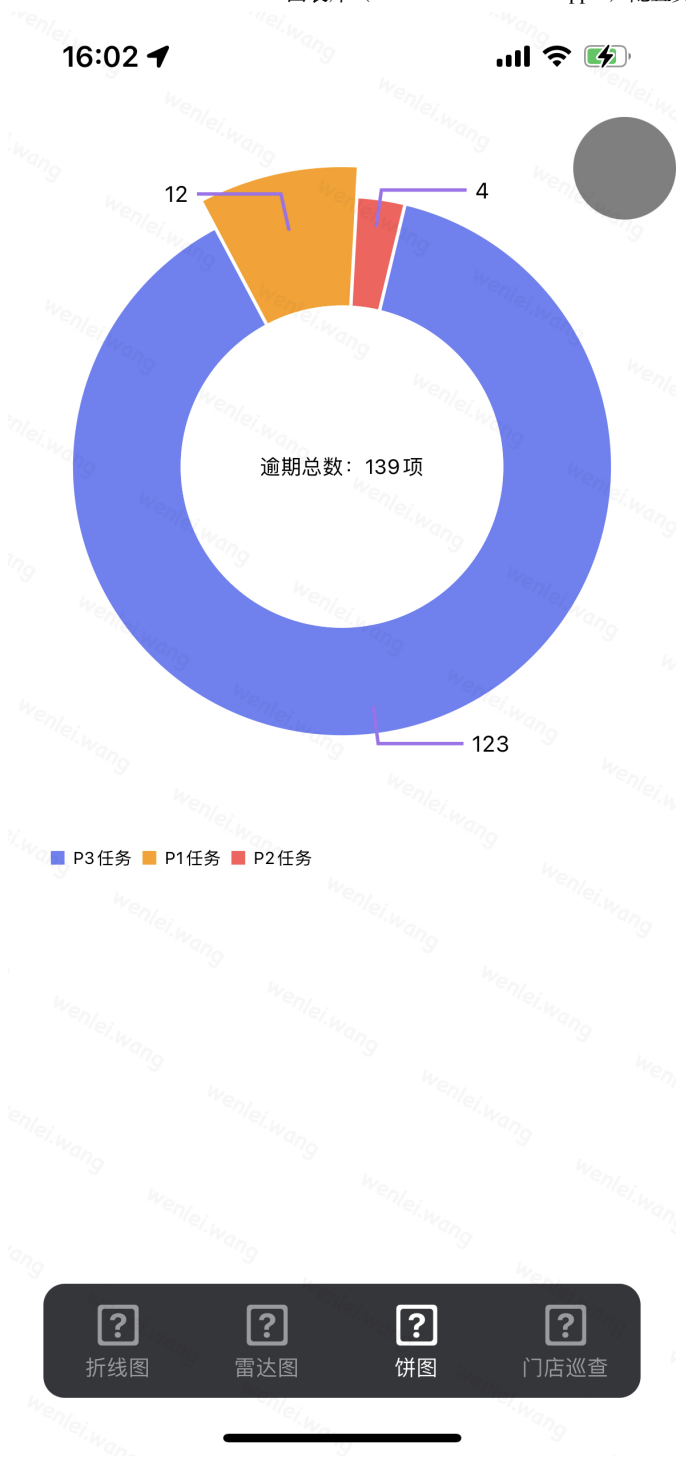
还原设计稿：

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项"
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
/>
```



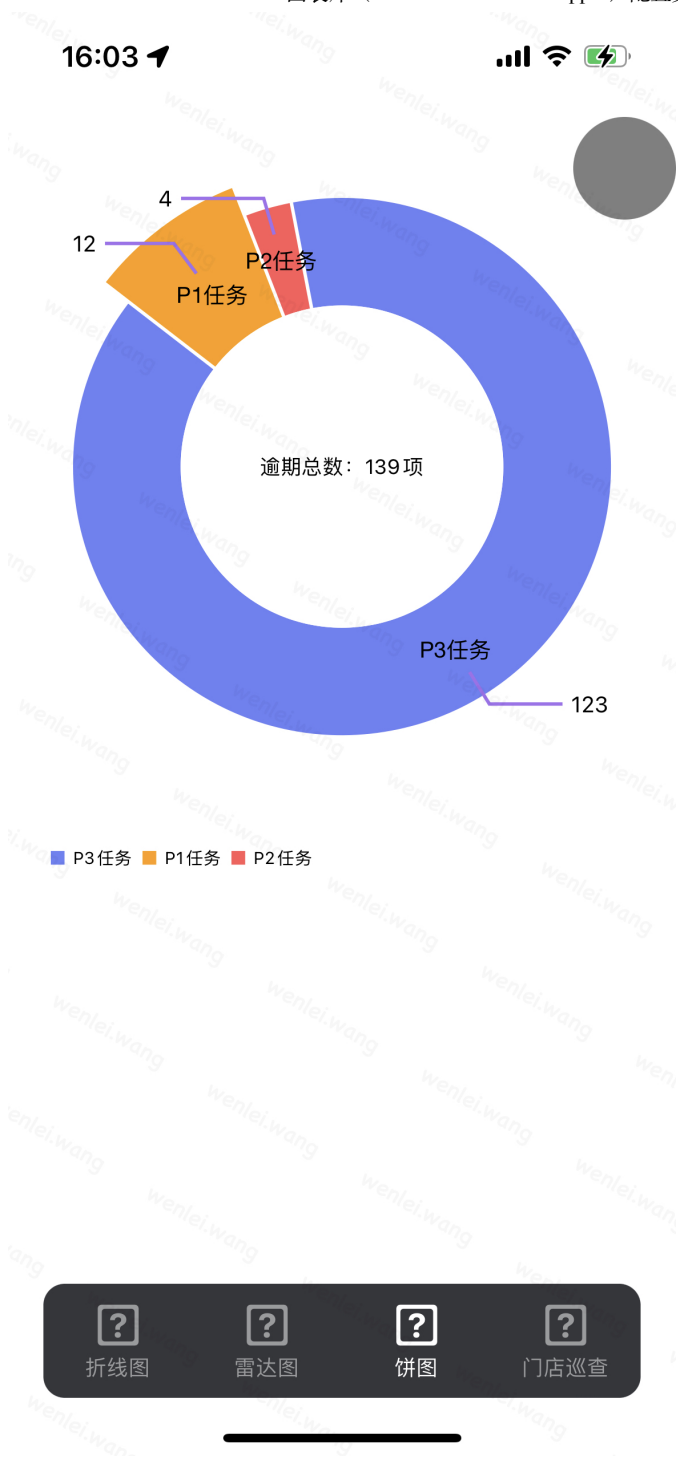
展示图例:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项"
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend // 增加此处
/>
```



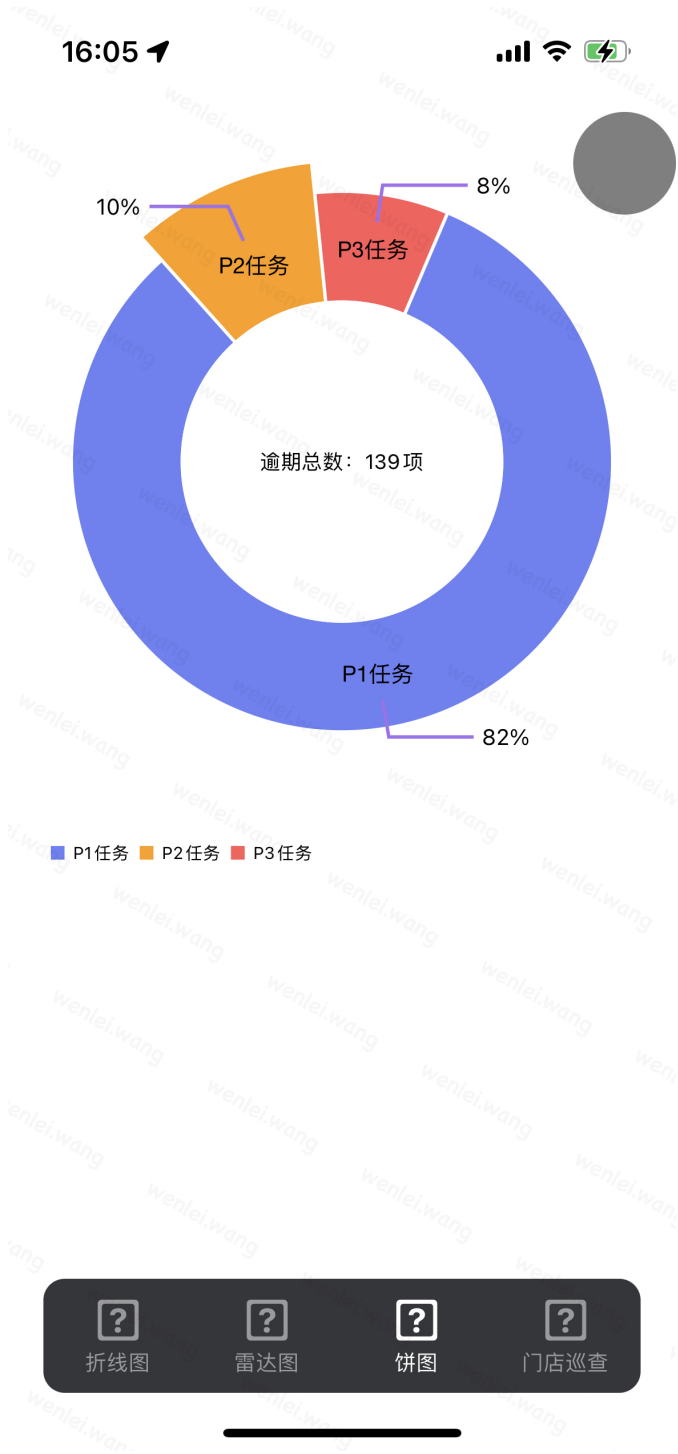
展示实体标签：

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项"
  showLabel={EValuePosition.INSIDE} // 增加此处
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
/>
```

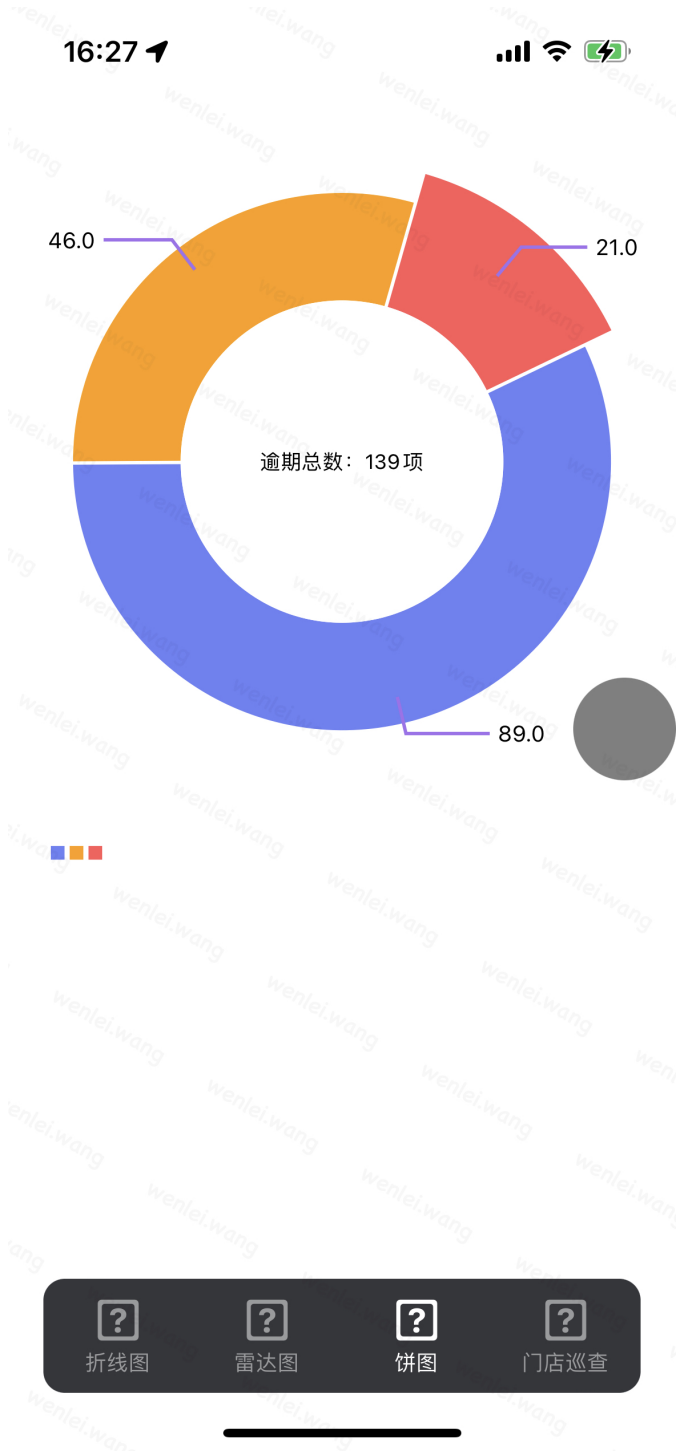
百分比格式:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={percentData} // 修改此处数据
  percentage // 增加此处
  centerText="逾期总数: 139项"
  showLabel={EValuePosition.INSIDE}
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
/>
```



简单数据格式：

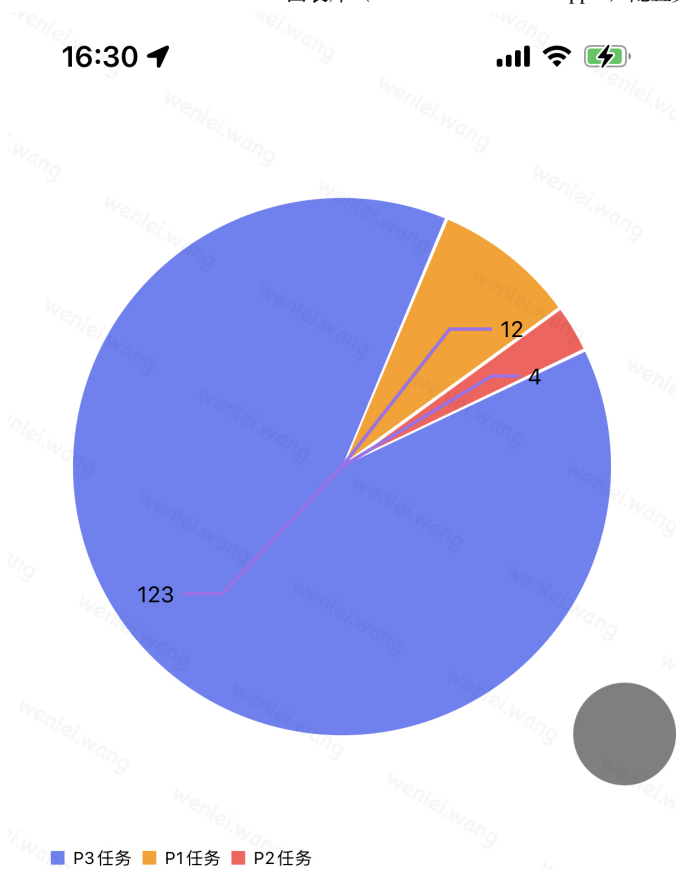
```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={simpleData} // 修改此处
  centerText="逾期总数: 139项"
  showLabel={EValuePosition.INSIDE} // 此时失效
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
/>
```



饼图模式:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项" // 此时失效
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
  usePie // 增加此处
/>
```

16:30



?

折线图

?

雷达图

?

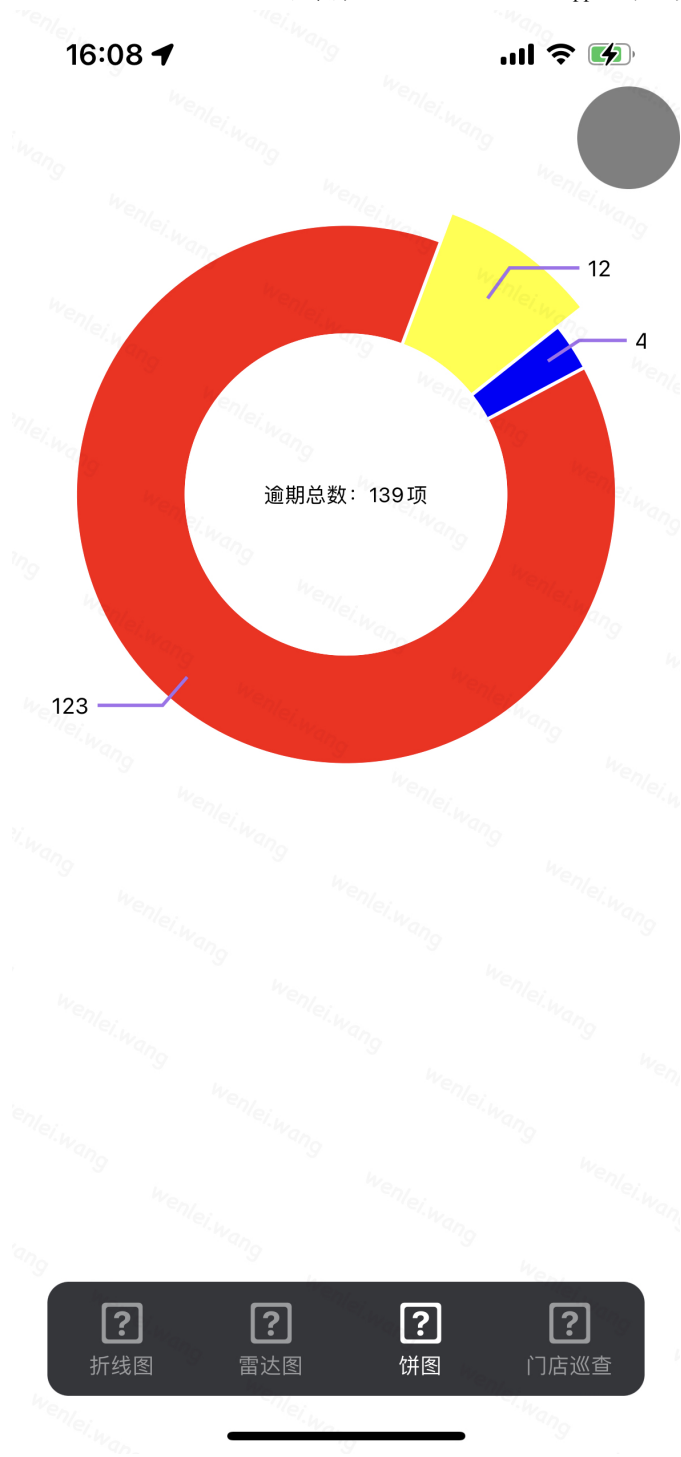
饼图

?

门店巡查

自定义主题颜色:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项"
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
  themeColors={[processColor('red'), processColor('yellow'), processColor('blue')]} // 增加此
/>
```



选择事件监听:

data为{ label: string, value: number }[]:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={testData}
  centerText="逾期总数: 139项"
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
  onSelect={ (data) => {
    if (typeof data === 'object') {
      console.log(data);
    }
  }} // 增加此处
/>
```

[Thu Apr 21 2022 16:08:55.533]	LOG	{"label": "P3任务", "value": 123}
[Thu Apr 21 2022 16:08:59.935]	LOG	{"label": "P1任务", "value": 12}
[Thu Apr 21 2022 16:09:00.932]	LOG	{"label": "P1任务", "value": 12}
[Thu Apr 21 2022 16:09:01.548]	LOG	{"label": "P2任务", "value": 4}

data为number[]:

```
<PieChart
  height={useWindowDimensions().height * 0.6}
  data={simpleData} // 修改此处
  centerText="逾期总数: 139项"
  showValue={EValuePosition.OUTSIDE}
  rotationEnabled={true}
  showLegend
  onSelect={ (value) => {console.log(value)}} // 修改此处
/>
```

[Thu Apr 21 2022 16:10:32.681]	LOG	3
[Thu Apr 21 2022 16:10:33.347]	LOG	4
[Thu Apr 21 2022 16:10:34.231]	LOG	4
[Thu Apr 21 2022 16:10:34.561]	LOG	1
[Thu Apr 21 2022 16:10:34.872]	LOG	2
[Thu Apr 21 2022 16:10:35.357]	LOG	3

Like Be the first to like this