# 第3章 图 表 区

图表区是图表的基本区域。所有的数据和图形都绘制在图表区中。从图形绘制范围来 分,图表区域分为外层图表区和绘图区。本章将详细讲解图表区的设置和创建。

# 3.1 图表区的构成

为了方便管理, Highcharts 将所有的图表元素都绘制在一个 Box 区域内,如图 3.1 所示。这个区域被称为图表区。由于图表中最重要的数据就是图表图形,所以图表图形所在的区域又构成一个独立的区域,称为绘图区,如图 3.1 所示。



图 3.1 图表区构成

从图 3.1 中可以看到,整个图表区由坐标轴构成的矩形分为两个部分:一部分是坐标 轴围成的绘图区,另一部分是图表区和绘图区之间的部分,称为外层图表区。下面依次讲 解这两个部分。

# 3.2 外层图表区

外层图表区是图表区和绘图区之间的部分。标题和图例这类图表元素往往都分布在这 个区域中。本节将详细讲解外层图表区的设置。

## 3.2.1 外层图表区的基本设置

外层图表区基本设置主要包括图表类型、图表区所在的HTML元素、外层图表区的宽和高等属性。下面依次讲解这些属性。

#### 1. 图表类型 type

在 Highchart 中,每种图表都是由 chart 组件实现的。所以,要得到不同类型的图表, 必须指定图表类型。图表类型由 type 配置项指定。语法形式如下:

type:string

其中, string 是 Highcharts 指定的图表类型字符串。其值可以为 area、arearange、 areaspline、areasplinerange、bar、goxplot、bubble、column、columnrange、errorbar、funnel、 gauge、heatmap、line、pie、pyramid、scatter、series、solidgauge、spline 和 waterfail。如 果不指定,默认就是 line 类型。

在第1章中的实例 1-1 中指定了图表类型为 line (折线) 类型。

type: 'line'

#### 2. 图表区所在 HTML 元素 renderTo

在脚本中,也可以通过配置项 renderTo 直接指定图表绘制在网页中的哪个容器中。语法形式如下:

renderTo: String

其中, String 是网页元素的 id。

在实例 1-1 中,直接使用了 jQuery 的选择器,所以没有指定该选项。

3. 指定外层图表区大小

在 Highcharts 中,可以通过配置项 width 和 height 指定外层图表区的宽和高。语法形 式如下:

width:Number1 height:Number2

其中,Number1和Number2分别是图表区的宽和高的值,单位为像素。

【**实例 3-1**: chartheightandwidth】修改实例 1-1 的代码,将图表区宽度设定为 300px, 高度设定为 200px。代码如下:

```
chart: {
   type: 'line',
   height: 200,
   width: 300,
   borderWidth: 1
}
```

//指定宽度 //指定高度

执行代码后,效果如图 3.2 所示。

• 41 •



图 3.2 设置图表区的长和宽

如果不设定图表区的长宽值, Highcharts 会根据图表区中的元素自动计算。如果图表 区包含的元素高度为 0,则将 Heigh 设置为 400px。

#### 4. 指定外层图表区背景颜色

为了美化图表显示效果,可以通过配置项 backgroundColor 来指定图表区的背景颜色。 语法如下:

backgroundColor: Color

其中, Color 是使用单引号括起来的颜色值, 默认值是#FFFFFF。

【**实例 3-2: chartbackgroundColor**】修改实例 3-1 的代码,设定图表区的背景颜色为 #FFFF00。代码如下:

```
chart: {
   type: 'line',
   height: 200,
   width: 400,
   borderWidth: 1,
   backgroundColor: '#FFFF00' //设定背景颜色
},
```

执行代码后,效果如图 3.3 所示。



图 3.3 设置图表区背景颜色

• 42 •

# 3.2.2 外层图表区的边框

在 Highcharts 中,允许为外层图表区设置边框。相关的配置项包括宽度 borderWidth、颜色 borderColor 和边框圆角半径 borderRadius。其语法如下:

```
borderWidth:Number1
borderColor:Color
borderRadius:Number2
```

其中,Number1表示边框的宽度,默认值为0,表示没有边框;Color表示颜色的字符串;Number2表示边框的圆角半径。Number1和Number2的单位均为像素px。

【**实例 3-3: chartborder**】下面修改实例 1-1 的代码,为图表区添加边框。需要修改的 代码如下:

执行代码后,效果如图 3.4 所示。



图 3.4 设置图表区的边框

# 3.2.3 外层图表区的范围

外层图表区位于图表区内部,绘图区外部,如图 3.5 所示的浅绿色部分所示。通常, 外层图表区的范围由 Highcharts 自动计算。



图 3.5 图表区和绘图区的间距

用户也可以通过 chart 组件的配置项 marginBottom、marginLeft、marginRight 和 marginTop 属性来分别设置两个区域的下边距、左边距、右边距和上边距。语法如下:

```
marginBottom:Number1
marginLeft:Number2
marginRight:Number3
marginTop:Number4
```

其中,Number1、Number2、Number3、Number4 分别表示各个边距值。单位为 px。 这4个配置项可以同时使用,也可以分开使用。

【实例 3-4: chartmargin】修改实例 1-1 的图表区和绘图区下间距,将其设置为 2。修改代码如下:

```
chart: {
  type: 'line',
  borderWidth: 1,
  marginBottom: 2, //设置下边距
  marginLeft: 30, //设置左边距
  marginRight: 30, //设置右边距
  marginTop:20 //设置上边距
},
```

执行代码后,效果如图 3.6 所示。从运行结果中可以看到,当边距设置不当时,会造成部分图表内容显示不完整,甚至丢失。如图 3.6 中,x 坐标轴的刻度没有显示。

为了简化操作, Highcharts 提供一个配置项 margin, 可以一次性对 4 个边距直接设置。 语法如下:

```
margin:Array
```

• 44 •



图 3.6 修改图表区和绘图区间距

其中, Array 是一个数组, 用来设置图表区和绘图区的上边距、右边距、下边距和左 边距。使用该属性后, 实例 3-4 的代码就可以简化为:

```
var options = {
    chart: {
        type: 'line',
        borderWidth: 1,
        margin:[2,30,30,20] //依次设置上边距、右边距、下边距、左边距
    },
```

# 3.2.4 外层图表区和图表内容的间距

为了美观,图表中的内容和图表区之间有一定的间距。如图 3.7 所示,绿色的部分为两者之间的间距。对于这部分间距,chart组件提供了对应的配置项来设置。例如,配置项 spacingBottom 用来设置下端间距;配置项 spacingLeft 用来设置左侧间距;配置项 spacingRight用来设置右侧间距;配置项 spacingTop 用来设置顶部间距。



图 3.7 图表区和图表内容的间距

这4个配置项的语法如下:

spacingBottom: Number1
spacingLeft: Number2
spacingRight: Number3
spacingTop: Number4

其中, Number1 参数用来设置底部间距, 默认值为 15; Number2 参数用来设置左侧间 距, 默认值为 10; Number3 参数用来设置右侧间距, 默认值为 10; Number4 参数用来设 置顶部间距, 默认值为 10。

【实例 3-5: spacing】下面取消实例 1-1 中图表内容和图表区的间距。修改代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   spacingBottom: 0, //设置底部间距
   spacingLeft: 0, //设置左侧间距
   spacingRight: 0, //设置右侧间距
   spacingTop:0 //设置顶部间距
},
```

执行代码后,效果如图 3.8 所示。从图中可以发现,取消间距后,图例和标题紧贴外 层图表区边框。



图 3.8 取消图表区和图表内容间距

为了方便设置, chart 组件提供了一个配置项 spacing, 用来同时设置 4 个间距。语法 形式如下:

spacing:Array

其中, Array 参数是由 4 个数值构成的数组。其中每个数值依次表示顶部、右侧、底部和左侧间距值。使用属性 spacing 可以简化实例 3-5 的代码。简化后的代码如下:

chart: {

• 46 •

```
type: 'line',
borderWidth: 1,
spacing:[0,0,0,0] //依次设置四个间距值
},
```

间距值也可以设置为负数,代码如下:

```
spacing:[-10,-10,-10,-10]
```

这个时候,会造成部分图表内容显示不完整,如图 3.9 所示。标题和图例都没有完全显示。



图 3.9 间距值为负值

△注意:版权信息部分不受间距影响。

# 3.2.5 外层图表区阴影

为了增加立体感, chart 组件提供阴影配置项。其语法如下:

shadow: Boolean|Object

其中, shadow 的参数可以是布尔值 true/false, 也可以是参数对象。 (1) 当参数为布尔值的时候,可以使用 true 和 false 表示是否有属性。 【**实例 3-6:** shadow】下面为实例 1-1 添加阴影效果。修改代码如下:

```
chart: {
   type: 'line',
   shadow:true
}
```

//启用阴影效果

},

执行代码后,效果如图 3.10 所示。在图表区的右侧和底端均出现灰色的阴影效果。

• 47 •



图 3.10 图表阴影

(2) 用户也可以采用对象的方式添加阴影效果。语法形式如下:

```
color:Color,
setX:Number1,
setY:Number2,
opacity:Number3,
width:Number4
```

{

其中,配置项 color 表示阴影的颜色;配置项 setX 表示阴影在 x 轴上的偏移距离;配置项 setY 表示阴影在 y 轴上的偏移距离;配置项 opacity 表示阴影的透明度;配置项 width 设置阴影的宽度。

【**实例 3-7: shadowObject**】使用对象的方式重新实现实例 3-6 的阴影效果。修改代码 如下:

```
chart: {
    type: 'line',
    shadow: {
        color: 'grey',
        //设置 x 轴方向的偏移距离
        offsetX: 1,
        //设置 y 轴方向的偏移距离
        opacity: 0.5,
        //设置透明度
        width: 2
        //设置阴影宽度
    }
},
```

## 3.2.6 图表区样式

为了便于用户制作出更美观的图表, chart 组件提供两个配置项, 用来设置图表所在容器 div 的 CSS 样式。下面依次讲解这两个属性。

• 48 •

#### 1. 内部样式 style

使用配置项 style,可以直接在 JavaScript 脚本中指定所使用的样式。其语法形式如下: style: CSSObject

其中,属性 CSSObject 是 CSS 对象,由 CSS 属性和值构成。其默认值如下:

{"fontFamily":"\"Lucida Grande\", \"Lucida Sans Unicode\", Verdana, Arial, Helvetica, sans-serif","fontSize":"12px"}

#### 2. 外部样式属性 className

为了方便用户管理 CSS 脚本, chart 组件提供了外部样式配置项 className。其语法 如下:

```
className: String
```

其中,参数 String 表示 CSS 类选择器名。具体使用方式,请参考 CSS 相关书籍。

# 3.3 绘图区

绘图区是 Highcharts 图表的最重要区域。所有的图表图形都绘制在该区域中。整个绘 图区由背景和边框两大部分组成。本节详细讲解绘图区的设置。

#### 3.3.1 绘图区的边框

在 Highcharts 的默认设置中,只有坐标轴 x 和 y 可以显示绘图区的左边界和下边界。为了 方便用户识别区域,也可以指定绘图区的边框。这样,可以更明确体现图表图形的范围。

#### 1. 绘图区边框宽度 plotBorderWidth

配置项 plotBorderWidth 用来设置绘图区的边框宽度。默认宽度为 0,所以在前面的实例中都看不到边框线。其语法如下:

plotBorderWidth:Number

其中,Number参数用来指定边框线的宽度,单位为 px。

【实例 3-8: plotBorderWidth】下面我们为实例 1-1 的图表绘图区添加宽度为 1px 的边框线。修改的代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   plotBorderWidth: 1
},
```

//设置边框线宽度为1

执行代码后,效果如图 3.11 所示。在图中,只标出了绘图区上边缘和右边缘的线。实际绘图区的左边缘和下边缘也有类似的边框线,只是被坐标轴覆盖,所以无法显示。当用

• 49 •

户设置的 borderWidth 值较大时,就可以显示出左侧边框和底端边框。



图 3.11 绘图区的边框线

#### 2. 边框线的颜色 plotBorderColor

从图 3.11 中可以看到,边框线已经显示出来,并且具有一定的颜色。绘图区边框线默认的颜色为#C0C0C0。开发者也可以使用配置项 plotBorderColor 设定。其语法形式如下:

```
plotBorderColor:Color
```

其中, Color 参数用来指定边框线的颜色。

【实例 3-9: plotBorderColor】修改实例 3-8 的代码,把绘图区边框的颜色设置为黑色。 修改代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   plotBorderWidth: 2,
   plotBorderColor:'#000000'
},
```

```
//设置边框线颜色为#000000
```

执行代码后,效果如图 3.12 所示。这里为了让边框线颜色更明显,将边框线的宽度设置为 2px。这时,就可以明显看出绘图区 4 个边缘的边框线。



图 3.12 设置为黑色的边框线

• 50 •

## 3.3.2 绘图区的背景

绘图区是图表图形展现的最重要的舞台。通过设置合理的背景,可以帮助用户更好地 识别数据。Highcharts 提供背景颜色和背景图案两种方式来美化绘图区。

#### 1. 绘图区背景颜色 plotBackgroundColor

为了明确绘图区的范围,不仅可以绘制边框线,还可以为绘图区设置背景颜色。这时 需要使用配置项 plotBackgroundColor。其语法形式如下:

```
plotBackgroundColor:Color
```

其中, Color 用来指定绘图区的颜色值。

【**实例 3-10:** plotBackgroundColor】修改实例 1-1,为绘图区添加灰色背景。修改代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   plotBackgroundColor:'#C0C0C0'
},
```

//设置绘图区背景色

执行代码后,效果如图 3.13 所示。从图中可以看到,绘图区已经被填充灰色。在这里, 还可以指定渐变色。我们将在后面的章节中进行讲解。



图 3.13 绘图区背景色

#### 2. 绘图区背景图片 plotBackgroundImage

在 chart 组件中,用户不仅可以通过设置背景色美化绘图区,还可以为绘图区设置背景图片。这时需要通过属性 plotBackgroundImage 设置。其语法如下:

plotBackgroundImage:String

其中, String 参数指定图片文件的 URL 地址。

【实例 3-11: plotBackgroundImage】修改实例 3-10,将绘图区的背景改成渐变色图 片 back.png。修改代码如下:

```
chart: {
    type: 'line',
    borderWidth: 1,
    plotBackgroundImage:'back.png' //设置背景图片
},
```

执行代码后,效果如图 3.14 所示。从图中可以发现,背景图片被拉伸直至和绘图区重合。所以,使用背景图片时需要注意背景图片变形问题,避免影响美观。



图 3.14 设置绘图区的背景图片

3.4 图表缩放

Highcharts 为图表提供缩放功能。为图表添加缩放功能后,当图表中的数据较多时,可以放大局部数据,便于用户查看数据细节。本节将讲解图表缩放功能的设置。

## 3.4.1 图表缩放类型

Highcharts 提供了 3 种缩放方式。这 3 种方式都是通过配置项 zoomType 来设置的。语法如下:

zoomType: String

其中,参数 String 指定缩放的方式。其值有以下几种。

□ None: 表示没有缩放, 这是默认值;

- □ x: 表示沿 x 轴方向缩放;
- □ y: 表示沿 y 轴方向缩放;

• 52 •

□ xy: 表示沿 x 轴和 y 轴同时缩放。

【实例 3-12: zoomType】下面为实例 1-1 的图表添加 x 轴方向的缩放。修改代码如下:

```
chart: {
   type: 'line',
   borderWidth:1,
   zoomType:'x' //设置 x 轴缩放
},
```

执行代码后,在绘图区中,使用鼠标沿着 x 轴方向拖动,会产生一个选择区域,如图 3.15 所示。松开鼠标后,图表会沿 x 轴方向放大,并多显示一个 Reset zoom 按钮,运行效 果如图 3.16 所示。当单击 Reset zoom 按钮时,将恢复原始的大小。



图 3.15 沿着 x 轴方向拖动鼠标



图 3.16 x 轴缩放后的图表

当参数 zoomType 设置为 y 或者 xy 后, 拖动的方式有所不同, 产生的选择区域也不同, 分别如图 3.17 和图 3.18 所示。







图 3.18 同时沿着 x 轴、y 轴方向拖动鼠标

# 3.4.2 填充选择区

在缩放的时候,为了便于用户识别选择的范围,选择区域会被自动填充。通过配置项 selectionMarkerFill,用户可以修改填充的颜色和透明度。语法如下:

selectionMarkerFill: Color

其中,参数 Color 指定填充颜色的 RGB 值和透明度。默认值为'rgba(69,114,167,0.25)'。 【实例 3-13: selectionMarkerFill】修改实例 3-12 的默认填充颜色的透明度。修改后的 代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   selectionMarkerFill: 'rgba(69,114,167,0.90)', //设定填充颜色
```

• 54 •

```
zoomType:'x'
},
```

执行代码后,效果如图 3.19 所示。从图中可以看出,填充颜色并没有发生变化,但透明度降低了。



图 3.19 设置填充颜色

## 3.4.3 重置按钮

当图表缩放后,图表会增加一个显示元素——Reset zoom 按钮,如图 3.20 所示。当用 户的鼠标移动到该按钮上,还会弹出一个提示框,提示用户单击该按钮后,将回到 1:1 的 比例。该按钮由 chart 的子组件 resetZoomButton 实现。下面详细介绍重置按钮的各项设置。



### 1. 本地化 Reset zoom 按钮

Reset zoom 按钮的名称和提示框内容都是英文的。这不太符合我们国内的需要。用户 需要将其本地化。Highcharts 提供一个本地化组件 lang 来实现基础元素的本地化工作。其 语法如下:

```
Highcharts.setOptions({
```

在这里, Reset zoom 按钮需要使用到两个配置项 resetZoom 和 resetZoomTitle。其中, resetZoom 用来设置按钮的文本内容; resetZoomTitle 用来设置提示框的文本内容。其语法如下:

```
resetZoom: String1
resetZoomTitle: String2
```

其中, String1 表示本地化的按钮文本内容; String2 表示本地化的提示框文本内容。 【实例 3-14: resetZoomlang】下面将 Reset zoom 按钮实现本地化。添加代码如下:

```
Highcharts.setOptions({
    lang: {
        resetZoom: '原始大小', //本地化按钮文本
        resetZoomTitle: '回到初始状态1:1' //本地化提示框文本
    }
});
```

执行代码后,效果如图 3.21 所示。



图 3.21 本地化 Reset zoom 按钮

#### 2. 设置位置 position

配置项 position 用来指定 Reset zoom 按钮显示的位置。其语法如下:

```
position:Object
```

其中,参数 Object 是一个位置对象,用来指定该按钮的对齐方式和位置坐标。该 Object 的结构如下所示:

```
align: String1,
verticalAlign:String2,
x:Number1,
y:Number2
```

- □ 配置项 align 用来指定水平对齐方式。String1 的值可以为 left、center 和 right, 默 认值为 right;
- □ 配置项 verticalAlign 用来指定垂直对齐方式。String2 的值可以为 top、middle 和 bottom, 默认值为 top;

• 56 •

- □ 配置项 x 用来指定水平偏移坐标值;
- □ 配置项 y 用来指定垂直偏移坐标值。

由于涉及坐标轴的确定和坐标偏移,所以这里详细讲解绘制过程中,按钮位置的计算 方式。主要包括以下3步。

(1) 根据配置项 align 确定 x 坐标轴的位置, 如图 3.22 所示。



图 3.22 水平对齐方式下的 x 轴位置

 ☐注意:根据对齐方式不同, x 轴的起始点不同。

(2) 根据配置项 verticalAlign 确定 y 坐标轴的位置,如图 3.23 所示。



图 3.23 垂直对齐方式下的 y 轴位置

▲注意:根据对齐方式不同,y轴的起始点不同。

(3)确定 x、y 坐标轴后,根据配置项 x、y 确定按钮的偏移位置。如果值为正的,就 顺着坐标轴方向偏移;否则,反向偏移。

对于{align:center,verticalAlign:middle,x:40,y:20}来说,其 Reset zoom 按钮位置如图 3.24 所示。



图 3.24 示例位置

【实例 3-15: resetZoomButton】下面不采用默认设置,重新设置 Reset zoom 按钮的 位置。修改代码如下:

```
chart: {
  type: 'line',
  borderWidth: 1,
  zoomType: 'x',
  resetZoomButton: {
    position: {
       align: 'left',
       verticalAlign: 'bottom',
       x: 20,
       y:-50
    }
}
```

},



图 3.25 修改 Reset zoom 按钮位置

## 3. 相对位置 relativeTo

设置按钮的对齐方式时,实际还涉及对齐方式所参考的标准。上面讲解的都是基于绘图区的。Highcharts 还允许用户基于图表区进行对齐。这时候需要使用配置项 relativeTo。 其语法如下:

```
relativeTo:String
```

其中,参数 String 用来指定参考标准。其值可以为'plot'、'chart'中的一个。默认是'plot'。 【实例 3-16: relativeTo】修改实例 3-15 的代码,将参考标准从默认的 plot,改为 chart。 修改代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   zoomType: 'x',
   resetZoomButton: {
```

• 58 •

```
relativeTo:'chart', //修改参考标准
position: {
    align: 'left',
    verticalAlign: 'bottom',
    x: 20,
    y:-50
    }
}
```

执行代码后,效果如图 3.26 所示。从图中可以明显看出, Reset zoom 按钮由原有的绘 图区的位置移动到绘图区的下边。



图 3.26 修改参考标准

#### 4. 设置按钮样式配置项 theme

用户除了可以设置复原按钮的位置,还可以进一步对按钮的样式进行设定。这时,需要使用属性 theme。其语法如下:

theme:Object

其中,参数 Object 用来设置按钮的各项属性。

## 3.4.4 缩放后的刻度计算

在沿着 x 轴缩放过程中, Highcharts 会根据显示区域的数值重新计算 y 轴的刻度。这 样便于浏览者更好地查看缩放后的数据,但也会让浏览者无法看到该部分数据相对其他数 据的趋势。为了避免这种情况发生, Highcharts 从 4.1.6 版本开始,在配置项 series 中增加 了一个新的配置项 getExtremeFromAll。其语法如下:

getExtremesFromAll: Boolean

该配置项的值为布尔类型。当值为 true 时, y 轴刻度将不会重新计算; 否则, y 轴刻

• 59 •

度会重新计算。默认值为 false。

【**实例 3-17: getExtremesFromAll**】下面设置缩放过程中 y 轴的计算方式。核心代码 如下:

执行代码后,效果如图 3.27 所示。从图中可以看出,缩放后 y 轴的刻度没有发生变化。



图 3.27 禁止 y 轴刻度重新计算

但是,如果将配置项 getExtremesFromAll 设置为 false 后,运行结果如图 3.28 所示。 这时 y 轴刻度是重新计算的。



图 3.28 允许 y 轴刻度重新计算

• 60 •

## 3.4.5 缩放扩展: 延 x 轴缩小

严格意义上讲,Highcharts 只提供了放大功能,并没有提供缩小功能。如果用户想缩 小坐标轴,只能通过重置按钮回到原来的大小,再重新缩放。该功能可以通过第三方插件 Zoom Out Selection 来实现,其下载地址为 https://github.com/RolandBanguiran/highstockzoom-out-selection/archive/master.zip。添加该插件后,用户可以从右向左选择区域时将缩小 显示。

【**实例 3-18: zoomoutx**】下面设置图表可以在 x 轴方向缩小。操作过程如下所述。 (1) 下载插件包,并解压。获取其中的脚本文件 zoomout-selection.js。

```
(2) 将该脚本文件加入网页中,代码如下:
```

```
<script src="zoomout-selection.js"></script>
```

```
(3) 启用 x 轴缩放功能。代码如下:
```

```
chart: {
    zoomType:'x'
```

```
z
```

执行代码后,从左向右框选区域,放大图表,效果如图 3.29 所示。当从右向左框选区域后,缩小图表,效果如图 3.30 所示。



图 3.29 放大图表



图 3.30 缩小图表

# 3.5 图表平移

当图表缩放后,图表往往会超出绘图区的显示范围,影响查看完整的数据。为了解决 这个问题,Highcharts提供了图表平移功能。下面将讲解如何使用平移功能。

## 3.5.1 启用图表平移

平移功能默认是禁用的。启用该功能,需要设置配置项 panning。其语法如下:

panning:Boolean

其中,参数 Boolean 是布尔值,确认是否启用平移功能。当其值为 true 时,启用该功能;当其值为 false 时,禁用该功能。默认值是 false。

### 3.5.2 设置图表平移方式

由于缩放和平移都是通过鼠标操作,所以需要设置一个按键来实现缩放和平移操作的 切换。这时候,需要使用配置项 panKey。其语法如下:

panKey:String

其中,参数 String 表示按键的名称。常用的按键名称包括'ctrl'、'alt'和'shift',分别表示 Ctrl、Alt 和 Shift 键。

【实例 3-19: panKey】为实例 3-16 添加图表平移功能。修改代码如下:

```
chart: {
   type: 'line',
   borderWidth: 1,
   zoomType: 'x',
                                        //启用平移功能
   panning: true,
                                        //设置功能切换键
   panKey:'ctrl',
   resetZoomButton: {
      relativeTo: 'chart',
      position: {
          align: 'left',
          verticalAlign: 'bottom',
          x: 20,
          y: -50
       }
   }
```

执行代码后,效果如图 3.31 所示。当用户按住 Ctrl 键并单击鼠标左键后,鼠标指针就 变成移动图表。这时,用户就可以左右移动图表,查看完整的图表数据。

### 3.5.3 设置延 y 轴平移

Highcharts 只提供了 x 轴的水平平移。当 y 轴的数据较多时,就无法通过平移的方式

• 62 •

进行浏览。这个时候,可以借助第三方插件 Y-Axis Panning 实现。该插件的下载地址为 https://github.com/RolandBanguiran/highstock-yaxis-panning/archive/master.zip。使用该插件 后,用户直接用鼠标就可以上下拖动图表。



图 3.31 启用平移功能

【实例 3-20: yaxispanning】下面为图表添加延 y 轴平移的功能。操作过程如下所述。 (1)下载该插件,并解压。获取其中的脚本文件 yaxis-panning.js。

(2) 在当前页面中引入该脚本。代码如下:

<script src="yaxis-panning.js"></script>

补充完整图表其他代码后,运行效果如图 3.32 所示。用鼠标向下拖动图表,效果如图 3.33 所示。



图 3.33 向下平移

5 --- Series 1 10

12.5

大学電

7.5

2.5

• 63 •