

目录

目录	1
表格	2
图表绘制	2
图表画法:	2
条件格式	2
指标选择器	3
聚合数据总计	3
一键排序	3
自定义排序	3
交叉表	3
对比表格	3
对比控件	4
自定义排序	4
仪表图	4
指标卡	5
图表画法:	5
指标看板	5
漏斗图	6
雷达图	6
类型一	6
图表画法:	6
类型二	6
图表画法:	6
类型三	6
图表画法:	7
桑基图	7
热力地图	7
旭日图	7
echarts热力地图	8
echarts标记地图	8
echarts轨迹地图	8
JavaScript自定义图表	9
echarts图表控件的实际应用: 3D地图	11
得分环	13
k线图	13

表格

图表绘制

以下将举例介绍表格如何绘制。比如：各个年度、各地区的销售额、利润、数量

将涉及到4个字段：维度“订单日期”，维度“地区”，度量“利润”，度量“数量”

图表画法：

1. 选择**表格**控件，在画布上画出**表格**。
2. 在表格的数据面板，列依次放置**维度订单日期**，**维度地区**，**度量利润**，**度量数量**。



表格支持在首列显示序号，设置方式如图所示。



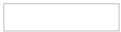
表格支持列宽的调整，选中表格后双击进入表格内操作，会显示辅助线，通过移动辅助线可以调整表格的列宽，如图所示。



表格图支持转置（即行列转换）



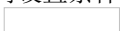
表格提供亚灰、简蓝、暗黑三种样式进行选择



条件格式

表格控件支持4种条件格式的设置：背景颜色、内容颜色、条形图、标记。4种条件格式的效果如图所示，可以叠加：

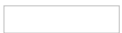


下面以“背景颜色”为例说明如何设置条件格式。第一步，右键选中需要设置条件格式的字段，在弹出的菜单中选择“条件格式”，再选择“背景颜色”。 

第二步，设置条件规则。



设置方式：可以选择“规则”或“色阶”，“规则”指仅有满足条件的单元格才会显示对应格式；“色阶”指在指定范围内的单元格会依据数值大小自动映射颜色深浅。

选择“规则”时的设置页面 

1、依据字段：当前字段可以依据其他字段的值来显示条件格式。例如地区字段可以依据销售额大小来显示条件格式，将销售额大于10000的地区字段标红。

2、显示范围：可以选择单元格、整行或序号。即设置条件格式应用的范围，是仅单元格，还是单元格所在的整行数据，或者单元格所在的序号字段（勾选显示序号后序号才会生效）。

3、设置条件：可以设置相应的条件规则。新增功能：

- 支持对某字段满足条件的前n项设置条件格式，例如将销售额最高的三个地区值标红（暂不支持表计算字段）。
- 可以与字段的第一行、平均值、最大值和最小值进行比较。
- 可以与其他字段进行比较，例如将成本大于销售额的值标红。
- 4、显示重复项：选择最高（前n项）或最低（前n项）时，在排序情况下可以设置是否显示重复项，例如，当某字段有7, 8, 9, 8四个值时，若设置了降序，选择最高前2项显示红色，则勾选显示重复项时9, 8, 8三个值显示红色，若不勾选，则9, 8两个值显示红色

选择“色阶”时的规则页面



1、依据字段：与“规则”一致。

- 2、空值处理：可以设置空值的显示颜色。
- 3、隐藏内容：勾选后会隐藏单元格内容（内容颜色无此选项）
- 4、设置条件：可以设置相应条件规则。新增支持自定义范围，即在指定范围内的单元格才会显示对应格式。

第三步，确定后规则生效，即可看到效果。

指标选择器

表格控件支持在浏览时动态选择需要展示的指标，效果如图所示。

1) 选择自定义指标 2) 勾选选择全部指标 3) 显示全部指标

设置方式非常简单，只需要在编辑界面，打开**自定义指标**设置即可。

打开**自定义指标**设置后，在设置弹窗中我们可以选择开放筛选的自定义指标，并设置默认选择项。

这里有几个需要明确的概念：

固定指标指的是固定于表格的指标，浏览报告时无法修改固定指标的显示状态；

自定义指标指的是浏览状态下允许用户自定义选择是否显示的指标，开启默认选中的自定义指标在初始状态下显示于表格。

聚合数据总计

平台的最新版本中对表格增强总计计算功能，可对视图已经聚合的数据进行平均、总计等计算。

一键排序

表格控件存在字段时，数据面板会出现设置按钮，选择设置按钮的排序级别，呼出排序级别弹窗，可在此对表格设置一键排序。

排序级别里提供【对所有未排序字段进行排序】功能，点击**升序**或**降序**后，将对所有字段都执行排序。执行过排序的字段将排列在下方，默认级别顺序按照数据面板上的字段顺序排列，也可对字段进行单独的排序设置。

自定义排序

表格的维度pill支持自定义排序，点击维度pill下拉箭头，选择**排序**。

在呼出的排序设置窗口中勾选自定义，允许对维度的各数据项进行自定义的置顶/置底排序。

交叉表

报告新增交叉表控件，可满足复杂报表的需求，如下图所示，选择交叉表控件，在列中拖入地区，行中拖入类别，同时拖入度量到指标，当拖入两个及以上度量时，列中自动添加度量名称pill，并在交叉表中显示度量名称，当指标中的度量被拖走只剩一个时，列中的度量名称pill也会自动消失。

和表格类似，可对交叉表每列指标设置不同条件格式：

对比表格

为了方便用户进行多维度数据的对比，新增对比表格控件。

1. 选择对比表格控件，在画布上画出对比表格。

2. 在数据面板拖入维度和指标。



3. 在对比设置中拖入维度，可以选择当选值和对比值进行对比，并选择显示的数据。



对比控件

对比表格可选择开启对比控件，对比控件的样式可自定义修改。



报告浏览者可随机切换对比项，灵活查看不同对比数据。



自定义排序

对比表格的维度pill支持自定义排序。



仪表图

在画布中央创建一个空白的仪表图，此时属性面板里没有添加任何数据内容，如图：



在当前值区域拖入一个度量字段，代表用户希望在在仪表图中显示的动态变化指标，此时整个仪表图范围都被该值覆盖，如图：



为了能够使图像表达的含义更丰富，我们需要给仪表图定一个范围，用户可以更直观的查看当前值与最大值之间的差距，这个范围即可以是固定的，即手动输入一个范围；也可以是动态变化的，即拖入两个度量字段，如图：



同时，用户如果不满意默认颜色，能够在样式面板中设定这两个区块各自的颜色，如图：



有时候用户需要设定多个目标，这时可以在目标值区域中拖入多个度量字段，最多能新增五个目标值；以当前图表举例，我需要设定去年同期的指标为一个目标值，即在目标值区域拖入（求和）去年同期字段，此时仪表图的指针指向代表当前值位置，红框区域代表去年同期目标值，可以清楚的对比出两者间的差距，如图：



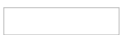
同样的，用户可以在样式面板对每一个区块的颜色进行设定，如图：



在有指针的情况下，用户可以自定义指针的颜色，如果选择跟随范围颜色，指针颜色会和范围颜色保持一致，如图：



如果该仪表图样式无法满足用户的需求，还可以在图表样式区域调整仪表图的样式和角度，如图：





指标卡

指标图用来显示关心的某一个指标值（即度量值）及其变化趋势（比如同比，环比）。

图表画法：

1. 选择**指标卡**控件，在画布上画出**指标卡**。
2. 将关心的字段（比如**销售额**）拖入指标区域。
3. 图标区域会显示指标数值，指标卡的左上角会显示指标的标题。



4. 在样式工具栏中可以调整指标的标题格式。

5. 如果需要同时显示指标的变化趋势（比如同比、环比），则需要将时间维度拖入日期范围区域，比如将“订单日期”拖入。 图片待添加

6. 拖入时间维度后，可以点击**设置**，弹出设置时间范围的页面。



7. 选定完时间范围后，可以勾选显示**同比**、**环比**。同比还可以选择**月同比**跟**月同比年同比**。值得注意的是，当选定的时间范围超过了一个月，则无法计算**月同比**。当选定的时间范围超过了一年，则无法计算**年同比**。



8. 另外可以在样式面板修改**同比**、**环比**的文案内容及颜色等样式。



9. 指标卡支持条件格式，下拉列表选择**条件格式**，呼出设置弹窗中可设置条件格式，同表格的条件格式设置，不再赘述。



指标看板

指标看板可以简单理解为多个指标卡的组合控件，帮助用户轻松实现添加多个指标卡。

指标看板的详细画法如下：

1. 选择**指标看板**控件，在画布上画出**指标卡看板**。



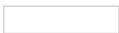
2. 将度量拖入**主要指标**，维度拖入**维度**，则生成最基础的指标看板，看板的指标显示各维度字段下的度量值。



3. 指标看板允许拖入多个主要指标，若开启次要指标，则将第二个及之后的主要主标自动拖入次要指标，看板中的指标卡依次显示主要指标的度量值和各个次要指标的度量值。



4. 在样式面板中可以分别对指标看板标题、整体样式、主要指标样式、次要指标样式、指标区域样式以及通用的边框圆角等设置。具体设置项与其他控件一致，此处不再赘述。



5. 指标看板的主要指标和次要指标均支持：筛选、数据格式、条件格式、空值处理、聚合方式、跳转等操作，具体规则与指标卡均保持一致。



6. 指标看板也允许添加同环比数据，且同环比的计算仅针对主要指标，设置规则与指标卡一致。增加同环比后，单指标卡的排列顺序依次为**主要指标**、**主要指标的同环比数据**、**次要指标**。



漏斗图

漏斗图示例如图所示



左侧是每个环节对应的指标值，右侧显示的是每个环节的转化率，不同颜色表示不同的环节。

图表画法1:

1. 选择“漏斗图”控件，在画布上插入**漏斗图**。
2. 将**环节**拖入维度区域，将**数量**拖入指标区域，即完成漏斗图的绘制。
3. 在样式面板，我们可以选择是否要显示转化环节名称和转化环节值。
4. 样式面板还可以选择颜色来区分不同的转化环节。

图表画法2:

1. 选择**漏斗图**控件，在画布上插入**漏斗图**。
2. 将多个度量拖入指标区域，即完成漏斗图的绘制。
3. 在样式面板，可以选择是否要显示转化环节名称和转化环节值。
4. 样式面板还可以选择颜色来区分不同的转化环节。

雷达图

平台支持三种类型的雷达图，下面将分别介绍绘制方式。

类型一



图表画法:

1. 选择**雷达图**控件，在画布上添加**雷达图**。
2. 选中**雷达图**控件，将1个维度字段拖入数据面板的**颜色**区域。
3. 将 ≥ 2 个度量字段拖入数据面板的**指标**区域。



类型二

图表画法:

1. 选择**雷达图**控件，在画布上添加**雷达图**。
2. 选中**雷达图**控件，将1个维度字段拖入数据面板的**外圈**区域。
3. 将 ≥ 2 个度量字段拖入数据面板的**指标**区域。



类型三



图表画法：

1. 选择**雷达图**控件，在画布上添加**雷达图**。
2. 选中**雷达图**控件，将1个维度字段拖入数据面板的**外圈**区域。
3. 将1个度量字段拖入数据面板的**指标**区域。
4. 再将1个维度子弹拖入数据面板的**颜色**区域。



桑基图

桑基图是一种流程图，适合特定场景下的数据可视化，比如要展示一段时间内不同国家之间的人口流动数据。

我们通过例子来进行演示，下图所示的数据，是一段时间内，各个国家之家的人口流动数（单位：万人）。



如果用桑基图来展示这份数据，我们可以更直观的看到不同国家之间的人口流动情况，如图所示。



该图由节点跟链接两种元素构成，每个节点表示一个国家，每个链接连接着两个节点，链接左边的是起始点（即人口流出国家），链接右边的是终止点（即人口流入国家），链接的粗细表示人口的多少。

热力地图

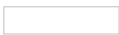
平台支持绘制热力地图，首先在画布上添加一个**热力地图**控件。



将需要的字段拖入控件的数据面板，即可绘制热力地图。如图所示，我们想分析杭州市的小区房价分布，则可将“小区经度”、“小区纬度”、“小区名称”、“小区均价”等字段拖入数据面板对应的区域。



双击热力地图，可以进入图表内操作，在该状态下，我们可以对热力地图进行放大并移动地图显示区域，以进一步了解关心的区域。



进入图表内操作后，鼠标悬停在数据点上时，会出现提示框显示数据点的详细信息，我们可以将需要显示的信息拖入数据面板的**提示信息**项。



旭日图

旭日图适合以层次结构展示数据构成，比如下图展示的是各地区、省份的销售占比。



可以把上述旭日图看作两个环图组合而成：



组合在一起以旭日图展示数据，不仅可以直观看到各地区销售占比、各省销售占比，还能看到地区跟省的层级对应关系。

那么在平台上如何绘制旭日图呢，非常简单。

1. 在画布上添加一个旭日图控件。



2. 将需要展示的字段拖入旭日图对应的区域，**地区、省**拖入维度区域，**销售额**拖入度量区域，即可完成旭日图的绘制。



另外，我们还可以在样式面板里对旭日图做一些样式上的调整，比如标签显示百分比。



echarts热力地图

平台接入了百度的echarts库，可以绘制echarts热力地图，地图背景为百度地图。热力地图适合于呈现地理分布信息，比如人口分布密度。

首先，在画布上添加一个echarts热力地图控件。



绘制echarts热力地图时支持两种方式：使用经纬度数据绘制、使用地址数据绘制。

使用经纬度数据绘制时，需要数据里提供了包含经纬度信息的数据字段。如图所示，我们将小区的经纬度、小区的居民户数等数据拖入对应的图表配置区域，即可完成人口分布热力地图的绘制。



使用地址数据绘制时，我们只需要提供地址信息，平台会解析地址后绘制图表，如图所示我们将包含小区地址、小区户数等数据字段拖入对应的图表配置区，平台会解析小区地址，完成热力地图的绘制。需要注意的是地址解析比较耗时，因此使用地址方式绘制地图时，建议数据项不宜过多，否则可能需要等待较长时间。



另外，鼠标移动至地图上的数据点时，会有提示框展示该数据点的一些详细信息，我们可以把需要展示的字段拖至**提示信息区域**。



echarts标记地图

平台接入了百度的echarts库，可以绘制echarts标记地图，地图背景为百度地图。

首先，在画布上添加一个echarts标记地图控件。



绘制echarts标记地图时支持两种方式：使用经纬度数据绘制、使用地址数据绘制。

使用经纬度数据绘制时，需要数据里提供了包含经纬度信息的数据字段，我们将经纬度数据拖入对应的配置区，即可在指定位置绘制标记点，如图所示，我们将杭州各小区的经纬度字段拖入图表配置区的**经度、纬度**区域，地图上即在对应的位置绘制出小区的标记点。



使用地址数据绘制时，我们只需要提供地址信息，平台会解析地址后在对应的位置绘制数据点，这里我们展示的是杭州滨江区的部分小区。需要注意的是地址解析比较耗时，因此使用地址方式绘制地图时，建议数据项不宜过多，否则可能需要等待较长时间。



另外，鼠标移动至地图上的数据点时，会有提示框展示该数据点的一些详细信息，我们可以把需要展示的字段拖至**提示信息区域**。



echarts轨迹地图

轨迹地图可以动态地展示物体（比如汽车）的运动轨迹，示例如图所示，显示的是北京地区某几条公交车的运行轨迹。



该份轨迹地图背后的数据如图所示，其中，bus_id字段用来区分不同的巴士，lng、lat字段存储的是巴士路线上经过的点的经纬度坐标，RecordTime则表示坐标的先后顺序。



我们将以上四个字段拖入轨迹地图配置面板对应的区域，即可绘制出每辆大巴（bus_id字段）在一段时间内（RecordTime字段）的位置（lng、lat字段）变化轨迹图。



JavaScript 自定义图表

平台支持用户编写JavaScript代码自定义绘制图表。

此功能仅平台旗舰版支持。

功能入口：[更多-echarts-JavaScript](#)



功能介绍：

第一步：插入echarts图表控件，在代码编辑区编写JavaScript代码，生成自定义图表。

与echarts官网在线编辑器类似，可直接在平台的代码编辑框中输入echarts配置项完成图表绘制，以基础的折线图为例，输入如下代码：

```
option = {
  xAxis: {
    type: 'category',
    data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
  },
  yAxis: {
    type: 'value'
  },
  series: [{
    data: [820, 932, 901, 934, 1290, 1330, 1320],
    type: 'line'
  }]
};
```

点击[应用](#)，代码运行无误，生成自定义图表：



代码中可以直接引用echarts变量，来调用echarts Api，也可对图表实例myChart变量来进行图表事件监听等操作。

以echarts的柱状图为例，输入如下代码：

```
option = {
  title : {
    text: '某地区蒸发量和降水量',
    subtext: '纯属虚构'
  },
  tooltip : {
    trigger: 'axis'
  },
  legend: {
    data:['蒸发量', '降水量']
  },
  toolbox: {
    show : true,
    feature : {
      dataView : {show: true, readOnly: false},
      magicType : {show: true, type: ['line', 'bar']},
      restore : {show: true},
      saveAsImage : {show: true}
    }
  },
  calculable : true,
  xAxis : [
```

```

    {
      type : 'category',
      data : ['1月', '2月', '3月', '4月', '5月', '6月', '7月', '8月', '9月', '10月', '11月', '12月']
    },
    yAxis : [
      {
        type : 'value'
      }
    ],
    series : [
      {
        name:'蒸发量',
        type:'bar',
        data:[2.0, 4.9, 7.0, 23.2, 25.6, 76.7, 135.6, 162.2, 32.6, 20.0, 6.4, 3.3],
        markPoint : {
          data : [
            {type : 'max', name: '最大值'},
            {type : 'min', name: '最小值'}
          ]
        },
        markLine : {
          data : [
            {type : 'average', name: '平均值'}
          ]
        },
      },
      {
        name:'降水量',
        type:'bar',
        data:[2.6, 5.9, 9.0, 26.4, 28.7, 70.7, 175.6, 182.2, 48.7, 18.8, 6.0, 2.3],
        markPoint : {
          data : [
            {name : '年最高', value : 182.2, xAxis: 7, yAxis: 183},
            {name : '年最低', value : 2.3, xAxis: 11, yAxis: 3}
          ]
        },
        markLine : {
          data : [
            {type : 'average', name : '平均值'}
          ]
        },
      },
    ]
  }
];

myChart.on('click', {seriesIndex: 1}, function() {
  myChart.setOption({
    series: [{}, {
      color: new echarts.graphic.LinearGradient(0, 0, 0, 1, [{
        offset: 0,
        color: 'rgb(55, 158, 168)'
      }, {
        offset: 1,
        color: 'rgb(55, 70, 131)'
      }])
    }])
});

```

点击应用，代码运行无误，生成自定义图表：



由于代码定义了图表标题，因此可将平台默认标题修改为不显示，拖拽调整整个图表的大小，即可得到一个好看的echarts图表：



第二步：拖入pill，引用pill生成的dataSet数据集。



平台的dataSet的默认数据格式为二维数组，以上述pill为例，dataSet值应为：

```
[[“东北”, 1000], [“西北”, 2000], ...]
```

可在代码编辑区输入：console.log(dataSet)，然后在浏览器控制台查看该数据集的完整数据。

用户可自由处理dataSet数据，并在配置项中引用：

```
// 向右侧数据面板中拖入字段，可在代码中引用dataSet字段获取二维数组数据
var data1 = dataSet.map(function(d) {return d[0];});
var data2 = dataSet.map(function(d) {return d[1];});
option = {
  xAxis: {
    type: 'category',
    data: data1
  },
  yAxis: {
    type: 'value'
  },
  series: [{
    type: 'line',
    data: data2
  }]
};
```

如无需对原始数据进行处理，可以直接在代码中用单独的数据集dataSet来声明数据，自由指定数据到视觉的映射：

```
dataset: { source: dataSet },
```

```
// 向右侧数据面板中拖入字段，可在代码中引用dataSet字段获取二维数组数据
option = {
  dataset: {
    source: dataSet
  },
  xAxis: {
    type: 'category',
  },
  yAxis: {
    type: 'value'
  },
  series: [{
    type: 'line'
  }]
};
```

第三步：边框样式的调整。

可拖拽改变图表大小，让其显示完全，也可在样式面板对标题及整个图表的背景、边框、圆角等作调整。

若代码运行有误，通过按 **Command+Option+I (Mac)** 或 **Control+Shift+I (Windows、Linux)**，打开 **Console** 面板，查看错误详情。代码编辑器默认底部吸附，点击红框按钮，可切换至自由移动的编辑框。若echarts实例中使用了jQuery请求内部数据或加载内部资源导致代码报错时，需要自行处理数据请求以及跨域问题。

echarts图表控件的实际应用：3D地图

将经纬度数据、想要展示的度量值，和地理坐标对应的城市名称维度拖入数据面板，然后利用外部请求获取地理位置信息，加上刚刚引入的数据，通过echarts gl绘制如下3D地图：

具体代码如下：

```
var barDataSet = dataSet.map(function (dataItem) {
  return [dataItem[0], dataItem[1], Math.sqrt(dataItem[2])];
});
var maxValue = Math.max(...barDataSet.map(d => d[2]));
fetch('https://ydcn.nosdn.127.net/echarts/assets/map/json/china.json').then(e => e.json()).then(chinaJson => {
  echarts.registerMap('china', chinaJson);
  myChart.setOption({
    backgroundColor: '#cdcfd5',
    geo3D: {
      map: 'china',
      shading: 'lambert',
      label: {
        show: true,
        textStyle: {
          color: '#000', //地图初始化区域字体颜色
          fontSize: 12,
        }
      }
    }
  });
});
```

```

        opacity: 1,
        backgroundColor: 'rgba(0,0,0,0)'
    },
    light: {
        main: {
            intensity: 5,
            shadow: true,
            shadowQuality: 'high',
            alpha: 30
        },
        ambient: {
            intensity: 0.5
        }
    },
    viewControl: {
        distance: 80,
        center: [0, -10, 0],
        panMouseButton: 'left',
        rotateMouseButton: 'right'
    },
    groundPlane: {
        show: true,
        color: '#999'
    },
    postEffect: {
        enable: true,
        bloom: {
            enable: false
        },
        SSAO: {
            radius: 1,
            intensity: 1,
            enable: true
        },
        depthOfField: {
            enable: false,
            focalRange: 10,
            blurRadius: 10,
            fstop: 1
        }
    },
    temporalSuperSampling: {
        enable: true
    },
    itemStyle: {
        color: '#fff',
        borderWidth: 1,
        borderColor: '#ccc',
    },
    regionHeight: 2
},
visualMap: {
    max: maxValue,
    calculable: true,
    realtime: false,
    inRange: {
        color: ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf', '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026']
    },
    outOfRange: {
        colorAlpha: 0
    }
},
series: [{
    type: 'bar3D',
    coordinateSystem: 'geo3D',
    shading: 'lambert',
    data: barDataSet,
    barSize: 0.5,
    minHeight: 0.2,
    silent: false,
    itemStyle: {
        color: 'orange'
    },
    emphasis: {
        label: {
            show: true,
            formatter: p => dataSet[p.dataIndex][3] + ':' + p.data[2]
        }
    }
}]
});

```

})

得分环

在画布中央创建一个空白的得分环，此时属性面板里没有添加任何数据内容，如图：

在当前值区域拖入多个度量字段，代表用户希望在得分环中显示的动态变化指标，此时得分环中显示所有指标的一半，如图：




为了能够使图像表达的含义更丰富，我们需要给得分环的每个指标分别定一个范围，用户可以更直观的查看当前值与最大值之间的差距，这个范围即可以是固定的，即手动输入一个范围；也可以是动态变化的，即拖入度量字段。

针对第一种情况，举例如下：

例如，对销售额的目标值为1000000，则将最大值设置为1000000，得分环中可以清晰的表明当前值与目标值的差距，如图所示：



针对第二种情况，举例如下：

例如，有时候用户需要与去年同期进行对比，则这时可以将度量字段拖入最大值，以另一个图表举例，需要设定去年同期的指标为一个目标值，即在最大值区域拖入“（求和）去年同期”字段，此时得分环的的最大值就是去年同期的数值，灰色部分代表去年同期目标值，可以清楚的对比出两者间的差距，如图：

同时，用户可以分别对每个指标设置最大最小值，得分环可以显示出每个指标与最大最小值的差距。



用户还可以在图表样式中选择顺时针或逆时针，如果不满意颜色或得分环的尺寸，可以对环底和环值分别进行设置。



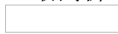


同样的，用户可以在样式面板对每一个区块的颜色进行设定，如图：



k线图

K线图是反映指标走势的一种图表，常用于展示股市及期货市场中价格信息和变化趋势，一根线段即反映了一天之内的指标变动情况，接下来将以股市数据为例，介绍如何绘制K线图。

首先在画布上添加一个“K线图”控件。

将日期、开盘价、收盘价、最高价与最低价分别拖至轴上的日期、起始值、终止值、最大值与最小值，将根据不同的日期粒度绘制不同粒度的K线图。 移动平均线指的是前x个统计周期（日、周、月、季度、年）的终止值的算术平均值，用户可以自行勾选是否显示MA5、MA10、MA20、MA30均线，也可以添加自定义的移动平均值。 用户可额外拖入一个度量，将以下方柱形图的方式进行展示。示例在此处拖入了交易量，以颜色对涨跌情况做了区分。 用户可拖动缩略轴上的时间范围，观察全局走势和详细内容。