



R软件在CGED-Q JSL中的运用

(四) 制图

作者：陈俊（硕士研究生，华中师范大学）；康文林（教授，香港科技大学；华中师范大学）





01

直方图

PART one

02

散点图

PART two

03

折线图

PART three

01

直方图



ggplot()函数

ggplot()函数的格式和逻辑:

```
ggplot(data,  
        mapping = aes(x,y,fill,color,...),
```

数据集

图层（美学），这里的aes函数中必须包含必要的X轴变量，和Y轴变量。

```
... ,  
)
```

一些调整图层的参数，没有固定的使用规则。

geom_bar()函数

geom_bar()函数的格式和逻辑:

ggplot(mapping = aes(x, y, fill, color,...),

data,

stat ,

position,

)

图层（美学），这里的aes函数中必须包含必要的X轴变量，和Y轴变量。

数据集，由于ggplot()函数已经设置，所以作图时geom_bar()中的data的默认值是Null。

stat参数有三种类型：其一是"count",表示计算频数；
其二是"identity",表示按照y的值进行计数
其三是"bin",表示对连续变量进行统计转换

position参数一共有3种形式

其一是"stack",堆积；
其二是"fill",且每个条形的最大值为1；
其三是"dodge",分条展示。

制图时，已建立好的图层函数`ggplot()`+制图函数`geom_bar()`函数即可制图：

实例代码：简单直方图

```
g1 <- ggplot(JSL1900_1912_only1910, aes(jiguansheng_sort))
```

```
g1+geom_bar(color="red", fill = "white", just= 0.5, width = 0.8)
```

进阶直方图:

第一步: 创建分析所需数据集

指定条件——只在兵部衙门任职的官员, 以该条件为基础生成数据集, 并做图

```
JSL1900_1912_bingbu <- subset(JSL1900_1912_delete_kongbaiming,机构 == "兵部衙門")
```

第二步：实例代码

简单直方图代码。其中，`geom_bar()`的
`position`参数，`position`参数一共有3种形式

#其一是"stack",堆积;
#其一是"fill",堆积,且每个条形的最大值为1;
#其一是"dodge",分条展示。

图标标题、x轴标题、y轴标题

```
g2 <- ggplot(JSL1900_1912_bingbu, aes(阳历年份numeric, fill = qiren))
g2+geom_bar(position = "dodge", just= 0.5, width = 0.5)+ labs(x = "年份", y = "记载数", title = "1900-1906年兵部的满汉比例")+
  guides(fill=guide_legend(title = "旗人"))+
  scale_fill_manual(values = c("cyan","pink"))+
  theme(plot.title = element_text(size = 15, hjust = 0.5, color = "orange"),
        axis.text.x = element_text(size = 9, angle = 30, color = "black", margin = margin(t = .5, unit = "cm")),
        axis.text.y = element_text(size = 9),
        axis.title.x = element_text(size = 10, hjust = 0.5, vjust = 0),
        axis.title.y = element_text(size = 10),
        legend.position = "bottom",
        legend.title = element_text(size = 12, hjust = 0.5, color = "orange" ),
        legend.background = element_rect(color = "black", fill = "grey90", linewidth = 1),
        legend.text = element_text(size = 10, color = "blue"),
        legend.key = element_rect(size = 2, color = "purple"),
        panel.background = element_rect(color = "grey50"),
        panel.grid = element_line(color="grey50", linewidth = 0.1))+
  scale_x_continuous(expand = c(0.1,0.1), breaks = seq(1900,1906,1))+
  scale_y_continuous(limits = c(0,1000), breaks = seq(0,1000,100))+
  geom_text(aes(label = ..count., family = "serif"), stat = "count", size = 3.5, vjust = -0.8, hjust = 0.5, color = "red2", position = position_dodge(0.5))
```

图例的标题

直方图各种细节的修饰, 后面有具体的theme () 函数的用法

x轴与y轴的最大最小值、间隔等

直方图数值标签的统计方式、大小、间隔、位置等

条形的颜色

直方图修饰函数theme()

<code>theme(plot.title = element_text(size = 15, hjust = 0.5, color = "orange"),</code>	----- 标题大小、位置、颜色
<code>axis.text.x = element_text(size = 9, angle = 30, color = "black", margin = margin(t = .5, unit = "cm")),</code>	-----x轴标签的大小、角度、颜色、标签与轴的间距
<code>axis.text.y = element_text(size = 9),</code>	----- y轴大小
<code>axis.title.x = element_text(size = 10, hjust = 0.5, vjust = 0),</code>	-----x轴标题的大小、与底部水平或垂直间距
<code>axis.title.y = element_text(size = 10),</code>	-----y轴标题大小
<code>legend.position = "bottom",</code>	-----图例位置（底部）
<code>legend.title = element_text(size = 12, hjust = 0.5, color = "orange"),</code>	-----图例标题大小、位置、颜色
<code>legend.background = element_rect(color = "black", fill = "grey90", linewidth = 1),</code>	-----图例背景的边框颜色、填充颜色、线条宽度
<code>legend.text = element_text(size = 10, color = "blue"),</code>	-----图例文本的大小、颜色
<code>legend.key = element_rect(size = 2, color = "purple"),</code>	-----图例节点的大小、颜色
<code>panel.background = element_rect(color = "grey50"),</code>	-----大背景的颜色
<code>panel.grid = element_line(color="grey50", linewidth = 0.1))</code>	-----大背景网格的颜色和线条大小

theme () 函数有非常多参数，这里提到的只是几个比较常用的，如果有更多需求，可以在R中help (theme) 查看可用的参数，或者登录ggplot的官网查询。值得注意的是，theme函数中的参数必须与element_text () 函数合用，element () 系列函数有4个：element_text () 、element_line () 、element_rect () 、element_blank () 。使用者必须根据参数的不同而调整element () 函数的使用，也可利用help功能查看具体用法。以下图片，比较形象的展示了theme () 函数中常用参数的用法，可供参考。

ggplot2 Theme Elements

`theme(element_name = element_function())`

- `element_text()`
- `element_line()`
- `element_rect()`
- `element_blank()`

Plot elements:

`plot.background`

`element_rect()`

`plot.title`

`element_text()`

`plot.margin`

`margin()`

Facetting elements:

`strip.background`

`element_rect()`

`panel.spacing`

`unit()`

`strip.text`

`element_text()`

Axis elements:

`axis.ticks`

`element_line()`

`axis.title`

`element_text()`

`axis.text`

`element_text()`

`axis.line`

`element_line()`

Legend elements:

`legend.margin`

`margin()`

`legend.title`

`element_text()`

`legend.key`

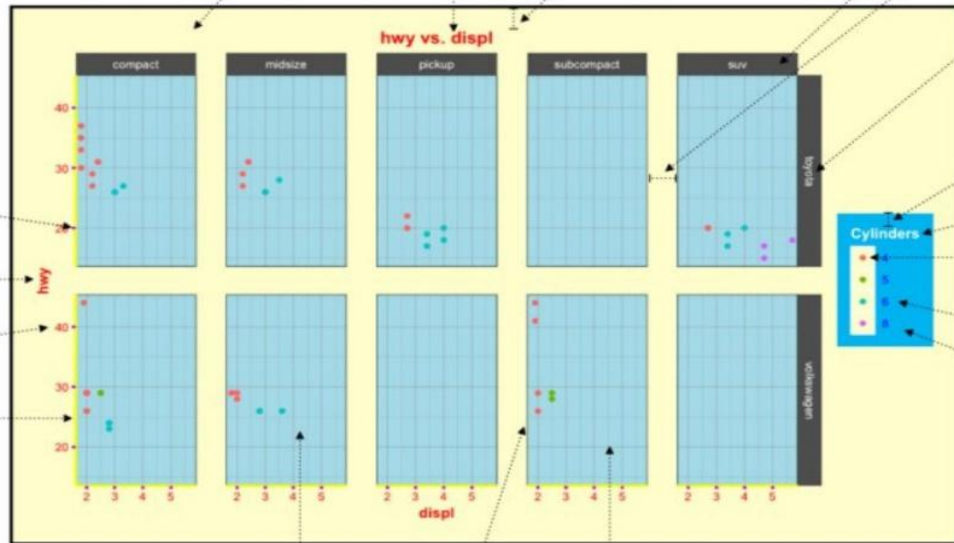
`element_rect()`

`legend.text`

`element_text()`

`legend.background`

`element_rect()`



`panel.background`

`element_rect()`

`panel.grid`

`element_line()`

`panel.border`

`element_rect(fill = NA)`

Panel elements:

henrywang.nl

Derived from "ggplot2: Elegant Graphics for Data Analysis"

<https://blog.csdn.net/zty0104>

图片来源

<https://blog.csdn.net/zty0104/article/details/119646934>

制定直方图标签的函数geom_text()

```
geom_text(  
  aes(label = ..count..  
  family = "serif"),  
  stat = "count",  
  size = 3.5,  
  vjust = -0.8,  
  hjust = 0.5,  
  color = "red2",  
  position = position_dodge(0.5)  
)
```

值标签字体
大小, 居中
格式, 颜色

值标签的统计方式, 这里选择
用..count.., 即表示让R统计频数
字体

stat参数有三种类型:
其一是"count",表示计算频数;
其二是"identity",表示按照y的值进行计数
其三是"bin",表示对连续变量进行统计转换

值标签的位置

1900-1906年兵部的满汉比例



图形效果

以上为进阶直方图制作的具体流程，即在简单直方图函数上加上主题修饰函数`theme ()`以及值标签制作函数`geom_text ()`、另外可根据制图需求选择加上调整横、纵坐标间隔的`scale ()`函数等。

01

散点图



散点图的制作流程和直方图类似，在建立图层的基础上套用`geom_point ()` 函数即可。

第一步：创建分析所用数据集

```
JSL1900_1912_bingbu$官职品级 <- ""  
尚书 <- str_detect(JSL1900_1912_bingbu$官职一,"尚書")  
JSL1900_1912_bingbu$官职品级[尚书 == TRUE] <- "尚书"  
左侍郎 <- str_detect(JSL1900_1912_bingbu$官职一,"左侍郎")  
JSL1900_1912_bingbu$官职品级[左侍郎 == TRUE] <- "左侍郎"  
右侍郎 <- str_detect(JSL1900_1912_bingbu$官职一,"右侍郎")  
JSL1900_1912_bingbu$官职品级[右侍郎 == TRUE] <- "右侍郎"  
主事 <- str_detect(JSL1900_1912_bingbu$官职一,"主事")  
JSL1900_1912_bingbu$官职品级[主事 == TRUE] <- "主事"  
員外郎 <- str_detect(JSL1900_1912_bingbu$官职一,"員外郎")  
JSL1900_1912_bingbu$官职品级[員外郎 == TRUE] <- "員外郎"  
郎中 <- str_detect(JSL1900_1912_bingbu$官职一,"郎中")  
JSL1900_1912_bingbu$官职品级[郎中 == TRUE] <- "郎中"  
七品京官 <- str_detect(JSL1900_1912_bingbu$官职一,"七品")  
JSL1900_1912_bingbu$官职品级[七品京官 == TRUE] <- "七品京官"  
JSL1900_1912_bingbu$品级 <- factor(JSL1900_1912_bingbu$官职品级, levels = c("尚书","左侍郎","右侍郎","郎中","員外郎","主事","七品京官"), labels =  
c(1.5,2,2,5,5.5,6,7))
```

创建一个新变量“官职品级”，赋值空白。

利用判定函数str_detect () 找出“官职一”为“尚書”的行，形成二元判定值 (T or F)。

二元判定值为“Ture”的行，给其“官职品级”一项赋值“尚书”，
以下几个官职的提取，同理。

利用factor () 函数将“官职品级”变量转化为表示品级的因子。

第二步：制作散点图

```
g3<- ggplot(data=subset(JSL1900_1912_bingbu, !is.na(品级)), aes(阳历年份,品级,shape = qiren,colour = qiren)) → 建立图层。  
g3+geom_point(position = "jitter")+
```

剔除缺失值

按所指定变量的不同类型区分形状

按所指定变量的不同类型区分颜色

```
labs(title = "1900-1906年兵部官员的品级及数量")+
```

```
guides(shape=guide_legend(title = "旗人"),colour=guide_legend(title = "旗人"))+
```

```
scale_colour_manual(values = c("gold","black"))+ → 点的颜色
```

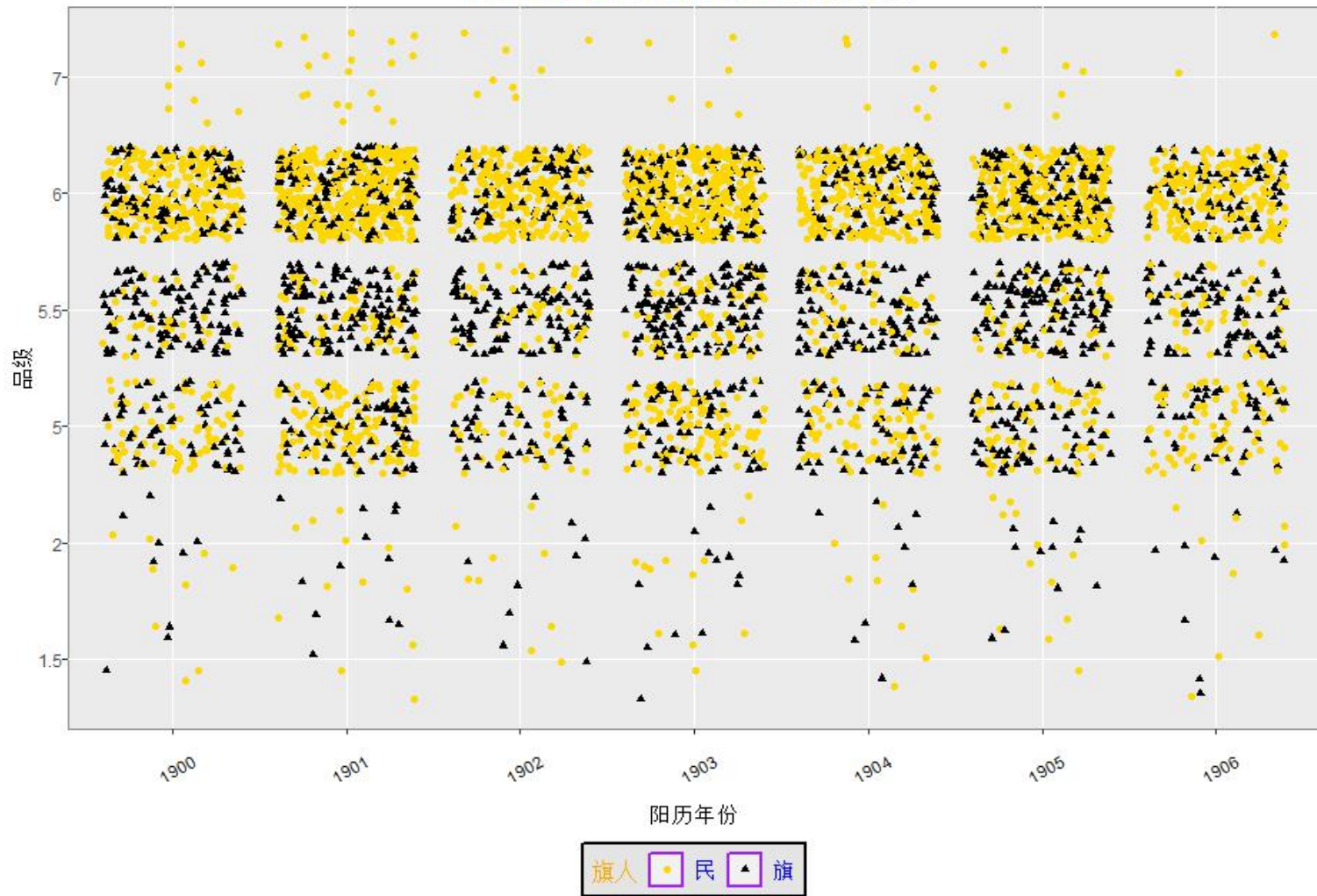
```
theme(plot.title = element_text(size = 15,hjust = 0.5,color = "orange"),  
axis.text.x = element_text(size = 9,angle = 30,color = "black",margin = margin(t = .5, unit = "cm")),  
axis.text.y = element_text(size = 9,margin = margin(t = .5, unit = "cm")),  
axis.title.x = element_text(size = 10,hjust = 0.5,vjust = 0),  
axis.title.y = element_text(size = 10),  
legend.position = "bottom",  
legend.title = element_text(size = 12,hjust = 0.5,color = "orange"),  
legend.background = element_rect(color = "black",fill = "grey90",linewidth = 1),  
legend.text = element_text(size = 10,color = "blue"),  
legend.key = element_rect(size = 1,color = "purple"),  
panel.background = element_rect(color = "grey50"),  
panel.grid = element_line(color="white",linewidth = 0.5))
```

散点抖动。由于设置的品级是连续型数值，如果不添加抖动，同一品级的所有观测值会集中在一个点上，会造成混淆，所以必须添加散点抖动。

theme () 函数

1900-1906年兵部官员的品级及数量

图形效果



01

折线图

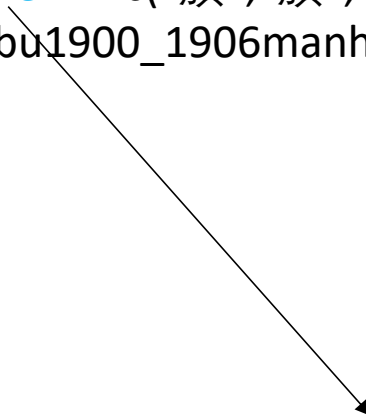


折线图的制作制作逻辑是：先创造散点图，形成点状图层；再制作折线图，将点状图层相连接，即形成折线图。

第一步：创建新的数据集

利用已知的1900-1906年兵部官员的满汉比例制作一个数据集

```
year <- c("1900","1901","1902","1903","1904","1905","1906")
percentage <- c(52.4,48.5,53.2,51.9,50.3,52.9,51.6,47.6,51.5,46.8,48.1,49.7,47.1,48.4)
shenfen <- c("旗","旗","旗","旗","旗","旗","旗","民","民","民","民","民","民","民")
bingbu1900_1906manhanbili <- data.frame(year,percentage,shenfen)
```



shenfen与percentage一一对应，方便后面制图

第二步：制作折线图

按变量类型对散点进行分组，如不分组，就会默认为全部点都在同一个组内，就不能进行比较。

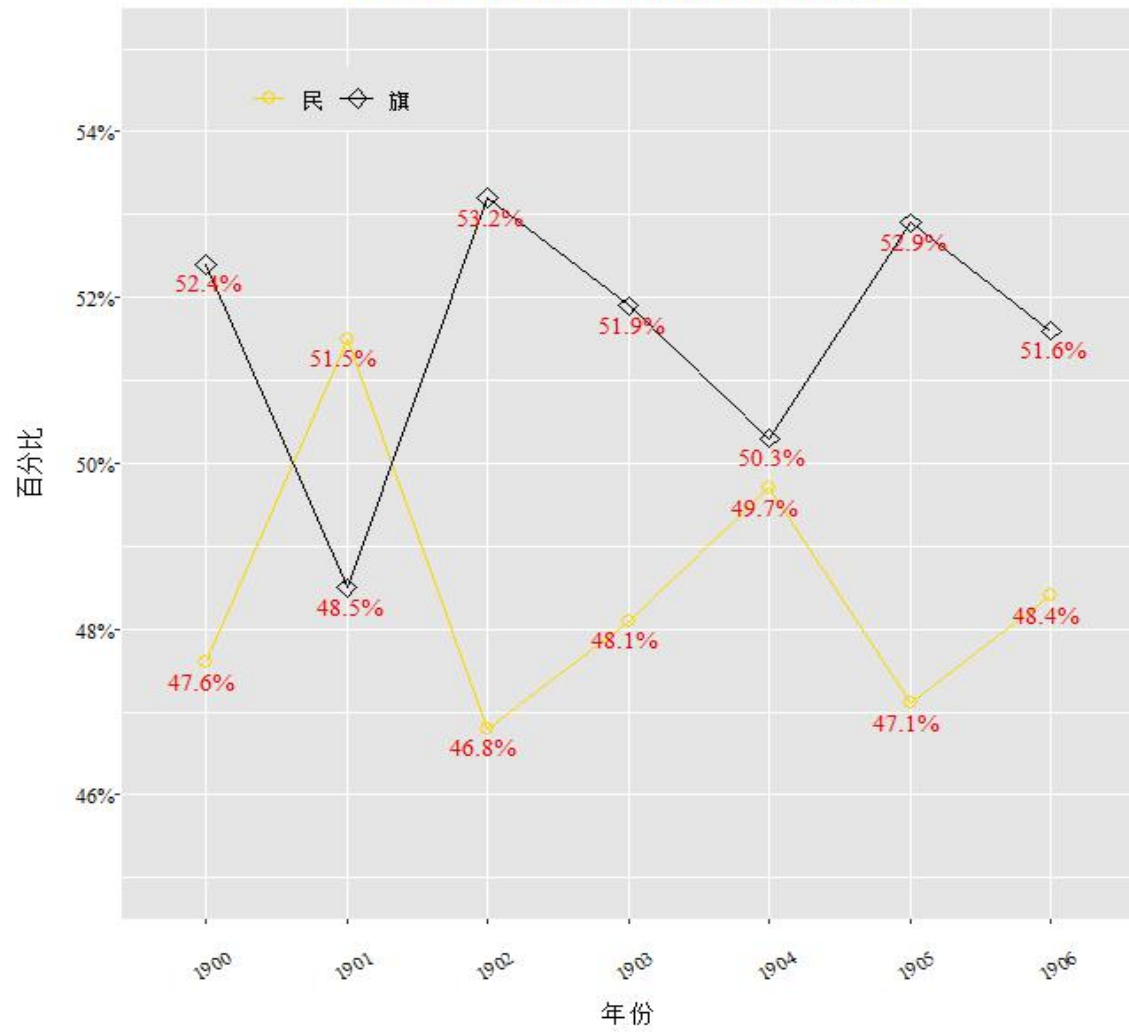
```
g5<- ggplot(bingbu1900_1906manhanbili,aes(x = year, y = percentage,group = shenfen,colour = shenfen,shape= shenfen))
g5+geom_point(size=2.5,stroke=0.5)+
geom_line()+
scale_color_manual(values = c('gold','black'))+
scale_shape_manual(values = c(1,5))+
labs(x="年份",y="百分比",title = "1900-1906兵部衙门满汉比例折线图")+
theme(legend.title = element_blank(),
      legend.text = element_text(family = 'serif'),
      legend.position = c(0.2,0.9),
      legend.direction = "horizontal",
      legend.background = element_rect(color="grey90",fill="grey90"),
      axis.text = element_text(color = 'black',family = 'serif'),
      axis.title = element_text(family = 'serif',size = 18,color = 'black'),
      plot.title = element_text(size = 15,hjust = 0.5,color = "orange"),
      axis.text.x = element_text(size = 9,angle = 30,color = "black",margin = margin(t = .5, unit = "cm")),
      axis.text.y = element_text(size = 9,margin = margin(t = .5, unit = "cm")),
      axis.title.x = element_text(size = 10,hjust = 0.5,vjust = 0),
      axis.title.y = element_text(size = 10,angle = 90,vjust = 0.5,margin = margin(0.1,0.5,0.1,0.5,'cm')),
      panel.background = element_rect(colour = "white",fill = "grey90"),
      legend.key = element_rect(color = "grey90",fill="grey90"))+
scale_y_continuous(limits = c(45,55),breaks = c(46,48,50,52,54),labels = c("46%","48%","50%","52%","54%"))+
geom_text(aes(label=paste0(percentage,"%")),family = "serif", stat = "identity",size = 4,vjust = 1.5,hjust = 0.5,color = "red2",position = position_dodge(0.1))
```

散点图层制作完毕后，加入线段进行连接

点的样式

这里用到了粘贴函数paste0 ()，将百分比与%符号粘贴在一起。

1900-1906兵部衙门满汉比例折线图



图形效果

以上即为三大基本图形的简单制作流程。R中的ggplot2包制作图形的方式如出一辙：首先建立图层，再在图层上加上需要建立的图形函数（geom_bar、geom_point、geom_line）、最后对建立的图形进行修饰并加上值标签。

谢谢

