

排座位计划-最终方案

杜丹琦 李昌骏

目录

1 任务概况	1
2 初步思路	3
3 想法的实现	7
4 最终的输出	13

1 任务概况

为了使同学们彼此之间更加熟悉，增强彼此之间的凝聚力，因此尝试在周五的统计软件与算法课上实行随机排座位。目前任务的概况如下：

统计软件与算法课一共有 92 名学生，其中研究生 70 名，本科生 22 名。

教室中的座位一共有 130 个，其具体分布如下表所示：

排数	左	中	右
第 1 排	3	6	3
第 2 排	4	6	4
第 3 排	4	6	4
第 4 排	4	6	4
第 5 排	4	7	4

排数	左	中	右
第 6 排	4	7	4
第 7 排	3	7	3
第 8 排	3	7	3
第 9 排	3	8	3
第 10 排	3	0	3
总计	35	60	35
合计	130		

可用下图表示：

```
# 加载相应的包
library(tidyverse)
library(ggplot2)

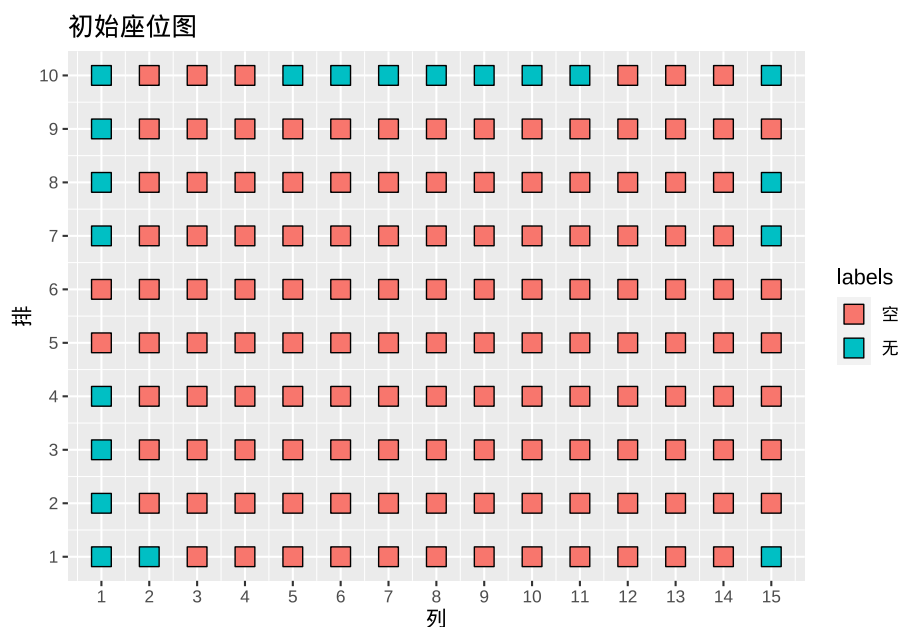
# 设置网格点
x_grid = seq(1,15)
y_grid = seq(1,10)
seat_grid = expand.grid(x_grid,y_grid)

# 给每个网格点加上标签
seat_grid = seat_grid %>% mutate(labels = " 空")

# 重命名列名
seat_grid = rename(seat_grid, col=Var1, row=Var2)

# 标记事实上没有位置的点
seat_grid[seat_grid$row==1&seat_grid$col %in% c(1,2,15),3] = " 无"
seat_grid[seat_grid$row %in% c(2,3,4)&seat_grid$col==1,3] = " 无"
seat_grid[seat_grid$row %in% c(7,8)&seat_grid$col %in% c(1,15),3] = " 无"
seat_grid[seat_grid$row==9&seat_grid$col==1,3] = " 无"
seat_grid[seat_grid$row==10&seat_grid$col %in% c(1,seq(5,11),15),3] = " 无"
```

```
# 画出图示
ggplot(seat_grid,aes(col,row,fill = labels)) +
  geom_point(shape = 22,size = 5) +
  scale_x_continuous(limits = c(1,15),breaks = seq(1,15,1)) +
  scale_y_continuous(limits = c(1,10),breaks = seq(1,10,1)) +
  labs(title = " 初始座位图",x = " 列", y = " 排")
```



2 初步思路

在进行座位分配时有以下几点考虑：

1. 统计软件与算法课需要带电脑，并且很多同学电脑的电池续航能力不足，需要自带充电器接通教室内的电源。但是，教室每一排电源数量是比容纳人数要少的，具体来说，每一区域每一排的电源数量，要比其容纳的人数少1个。如中间区域第一排有6个座位，但是电源只有5个。同时考虑到若每一排都坐满，同学们会感到比较拥挤（这一点对中间一排比较明显）。根据本周二晚上在应用多元统计分析课程（该课程的选课人数与选课结构均与

统计软件与算法课程相似)上的观察: 同学们自发形成的座位格局中, 中间区域没有一排坐满, 而两旁的区域也只有共计大概三排左右坐满。

因此决定在**每一区域的每一排**都空出一个座位, 既是为了保证电源数量, 也是为了让同学们入座时可以自由调整, 坐得更加舒服。对于两侧的座位, 由于最边缘的位置看黑板不太方便, 因此将两侧最边缘的位置空出; 而对于中间位置来说, 由于靠中的位置进出不太方便, 因此将居中的位置空出; 以及, 两侧位置的第 10 排太过遥远, 因此直接不坐人。这样一来, 座位还剩下:

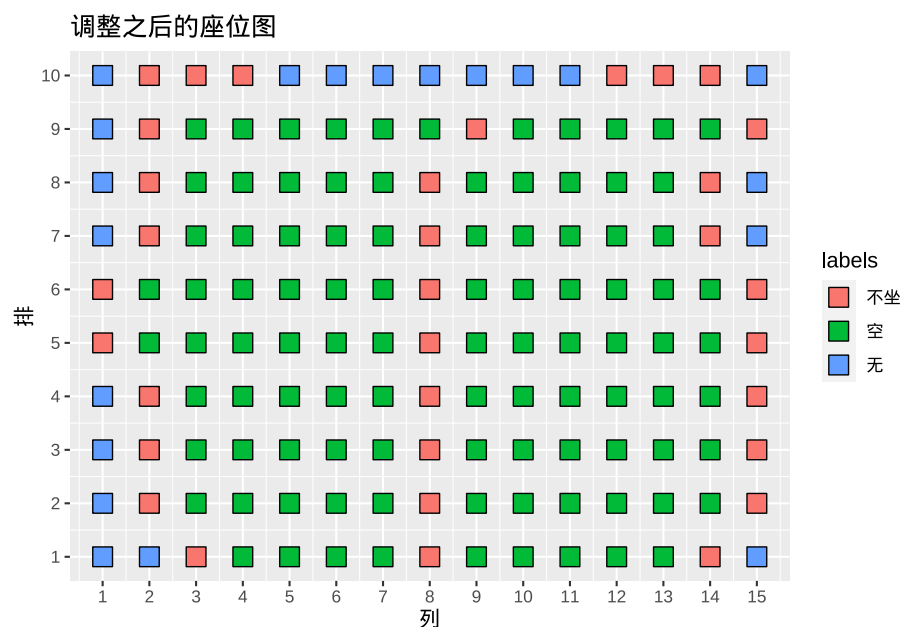
$$130 - 2 \times 3 - 3 \times 9 = 97$$

还剩下 97 个位置, 比起 92 名同学, 还是要多的, 故座位足够。

接下来可视化选择之后的座位图:

```
# 设置不坐的座位图
seat_grid[seat_grid$row == 10 & seat_grid$col %in% c(2,3,4,12,13,14),3] = " 不坐"
seat_grid[seat_grid$row %in% seq(1,8) & seat_grid$col == 8,3] = " 不坐"
seat_grid[seat_grid$row == 9 & seat_grid$col == 9,3] = " 不坐"
seat_grid[seat_grid$row %in% c(5,6) & seat_grid$col %in% c(1,15),3] = " 不坐"
seat_grid[seat_grid$row %in% c(2,3,4,7,8,9) & seat_grid$col %in% c(2),3] = " 不坐"
seat_grid[seat_grid$row %in% c(2,3,4,9) & seat_grid$col %in% c(15),3] = " 不坐"
seat_grid[seat_grid$row %in% c(7,8) & seat_grid$col %in% c(14),3] = " 不坐"
seat_grid[seat_grid$row %in% c(1) & seat_grid$col %in% c(3,14),3] = " 不坐"

# 画出图示
ggplot(seat_grid, aes(col, row, fill = labels)) +
  geom_point(shape = 22, size = 5) +
  scale_x_continuous(limits = c(1,15), breaks = seq(1,15,1)) +
  scale_y_continuous(limits = c(1,10), breaks = seq(1,10,1)) +
  labs(title = " 调整之后的座位图", x = " 列", y = " 排")
```



2. 统计软件与算法这门课中有研究生，也有本科生。但是本学期时间只剩下四周，并且考虑到研究生同学与本科生同学相互认识的意愿较弱，因此将研究生和本科生分开来坐。大致设想为：

在之前调整之后的座位图中，两侧座位能够容纳 23 人，大致与本科生数量相当，而中间的座位，靠左的一半能够容纳 24 人，靠右的一半能够容纳 27 人，均与本科生人数相当，因此接下来的四次课，可考虑每次随机将本科生放到教室左侧、右侧、中间靠左、中间靠右。

为了方便后面的抽样，给每一个位置增添区域特征：

```
# 增添区域特征
```

```
seat_grid = seat_grid %>% mutate(position = "无")
```

```
# 设置各个座位的区域
```

```
# 设置左方区域
```

```
seat_grid[seat_grid$row==1&seat_grid$col %in% c(3,4,5),4] = "左"
```

```
seat_grid[seat_grid$row %in% c(2,3,4)&seat_grid$col %in% c(2,3,4,5),4] = "左"
```

```
seat_grid[seat_grid$row %in% c(5,6)&seat_grid$col %in% c(1,2,3,4),4] = "左"
```

```

seat_grid[seat_grid$row %in% c(7,8,9,10)&seat_grid$col %in% c(2,3,4),4] = " 左"

# 设置右方区域
seat_grid[seat_grid$row==1&seat_grid$col %in% c(12,13,14),4] = " 右"
seat_grid[seat_grid$row %in% c(2,3,4,5,6)&seat_grid$col %in% c(12,13,14,15),4] = " 右"
seat_grid[seat_grid$row %in% c(7,8)&seat_grid$col %in% c(12,13,14),4] = " 右"
seat_grid[seat_grid$row %in% c(9)&seat_grid$col %in% c(13,14,15),4] = " 右"
seat_grid[seat_grid$row %in% c(10)&seat_grid$col %in% c(12,13,14),4] = " 右"

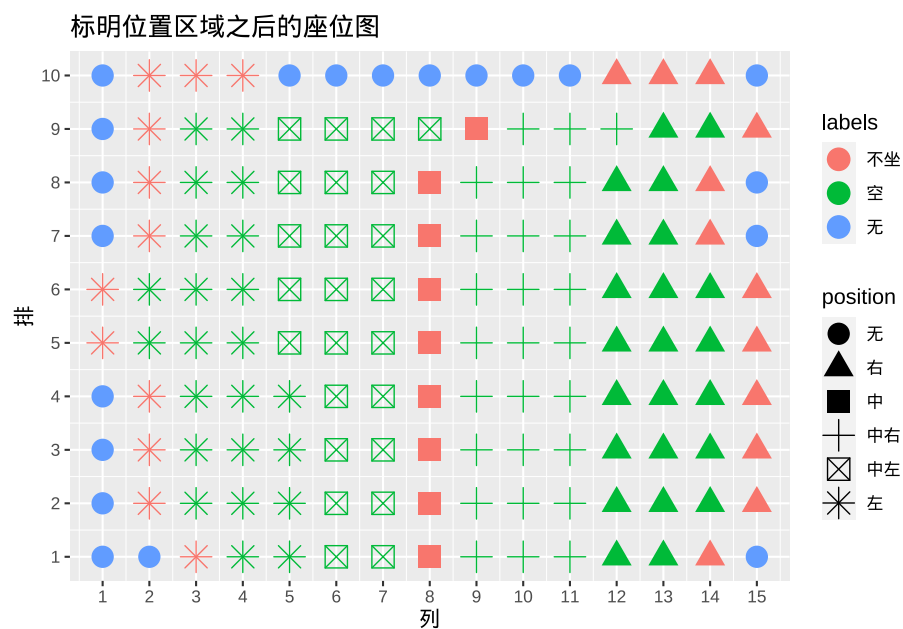
# 设置中间靠左的区域
seat_grid[seat_grid$row %in% c(1,2,3,4)&seat_grid$col %in% seq(6,7),4] = " 中左"
seat_grid[seat_grid$row %in% c(5,6,7,8) &seat_grid$col %in% seq(5,7),4] = " 中左"
seat_grid[seat_grid$row %in% c(9) &seat_grid$col %in% seq(5,8),4] = " 中左"

# 设置中间靠右的区域
seat_grid[seat_grid$row %in% seq(1,8)&seat_grid$col %in% seq(9,11),4] = " 中右"
seat_grid[seat_grid$row == 9&seat_grid$col %in% seq(10,12),4] = " 中右"

# 将中间不坐的位置设置为中
seat_grid[seat_grid$row %in% seq(1,8)&seat_grid$col == 8 ,4] = " 中"
seat_grid[seat_grid$row == 9&seat_grid$col == 9 ,4] = " 中"

# 可视化
ggplot(seat_grid,aes(col,row,color = labels,shape = position)) +
  geom_point(size = 5) +
  scale_x_continuous(limits = c(1,15),breaks = seq(1,15,1)) +
  scale_y_continuous(limits = c(1,10),breaks = seq(1,10,1)) +
  labs(title = " 标明位置区域之后的座位图",x = " 列", y = " 排")

```



3 想法的实现

首先载入相应的数据：

```
# 载入相应的包
library(readxl)
library(tidyverse)

# 载入所有人名单

# 载入本科生名单
undergraduate = read_excel("C:/Users/Hasee/Desktop/本科生名单.xlsx", col_names = FALSE)

## New names:
## * `` -> `...1`

# 标注年级为本科生
undergraduate = undergraduate %>% mutate(" 年级" = " 本科生")
```

```
# 载入研究生名单
postgraduate = read_excel("C:/Users/Hasee/Desktop/研究生名单.xlsx", col_names = FALSE)

## New names:
## * `` -> `...1`

# 标注年级为研究生
postgraduate = postgraduate %>% mutate(" 年级" = " 研究生")

# 重新标注列名
undergraduate = undergraduate %>% rename(" 姓名" = "...1")
postgraduate = postgraduate %>% rename(" 姓名" = "...1")

# 整合数据，合成一个学生名单
name_list = rbind(undergraduate, postgraduate)
```

载入之后统计一下人数，看人数是否对得上：

```
# 统计结果，看看人数是否对得上
name_list %>% group_by(年级) %>% summarise(n = n())

## # A tibble: 2 x 2
##   年级      n
##   <chr> <int>
## 1 本科生    22
## 2 研究生    70
```

发现人数正确，可以进行下一步操作。

```
# 按照年级进行排序
name_list = name_list %>% arrange(年级)
```

构造一个专门用于抽样的抽样框，在原数据框的基础上，添加“name”一列方便后续填入名字。

随机分座的效果是通过对座位数据框随机排序之后和姓名数据框一一配对

来实现。

```
# 构造用于抽样数据框
sample_data_frame = seat_grid

# 增加姓名一列
sample_data_frame = sample_data_frame %>% mutate(name = "空")

# 选取所有状态为“空”和“不可坐”的座位
sample_data_frame = sample_data_frame[!sample_data_frame$labels == "无",]

# 设置将要从中进行抽样的向量

# 座位的序列向量
n_sample = nrow(sample_data_frame)

# 设置随机种子，确保结果可以复现
set.seed(128)

# 进行抽样

# 对座位数据框的行数进行随机抽样，并以此更新座位数据框，相当于随机改变座位数据框的顺序
sample_order = sample(seq(1:n_sample),size = n_sample,replace = FALSE)
sample_data_frame = sample_data_frame[sample_order,]

# 进行第一轮迭代，将本科生的位置排好，以本科生坐在教室的左侧为例

i = 1
for(j in 1:130){
  # 设置条件语句，使得本科生全部被分配在教室的左侧位置
  if(sample_data_frame[j,"position"] == "左"&sample_data_frame[j,"labels"] == "空"){
    sample_data_frame[j,"name"] = name_list[[i,1]]
    i = i + 1
  }
  # 安排好本科生之后，跳出此次循环
```

```
    if(i > 22){
      break
    }
  }

# 进行第二轮迭代，将研究生的位置排好
i = 23
for(j in 1:130){
  if(sample_data_frame[j,"name"] == "空"&sample_data_frame[j,"labels"] == "空"&!sample_data_frame[j,"position"] == "空"){
    sample_data_frame[j,"name"] = name_list[[i,1]]
    i = i + 1
  }
  # 当全班所有人都被安排好座位之后，结束迭代
  if(i > 92){
    break
  }
}
```

接下来进行数据可视化：

```
# 设置用于储存画图数据的数据框
seat_plot = data.frame(col = 0 ,
                        row = 0,
                        labels = 0 ,
                        position = 0,
                        name = 0)

# 由于绘图时不需要标注“无”，因此重新放置每个座位的位置
for(i in 1:9){
  # 载入每一排的数据
  seat_row = sample_data_frame[sample_data_frame$row==i,]
  # 按照列的大小对每一排的位置进行排序
  seat_row = arrange(seat_row,col)
  n = nrow(seat_row)
```

```

row_name = data.frame(row = c(i),
                      col = seq(1,15,length.out = n),
                      labels = seat_row[,3],
                      position = seat_row[,4],
                      name = seat_row[,5])

# 对每一排的数据进行整合
seat_plot = rbind(seat_plot,row_name)
}

# 删除不用的第一行数据
seat_plot = seat_plot[-1,]

# 将“中左”和“中右”位置统一改为“中”
seat_plot[seat_plot$position==" 中左"|seat_plot$position==" 中右",4] = " 中"

# 将之前设定“不坐”的位置恢复为“空”
seat_plot[seat_plot$labels==" 不坐", "labels"] = " 空"

# 设置用于标注名字的数据框
seat_name = seat_plot

# 将标注名字的位置的纵坐标加上 0.5, 使得名字与座位图案分开
seat_name[,2] = seat_name[,2]+0.5

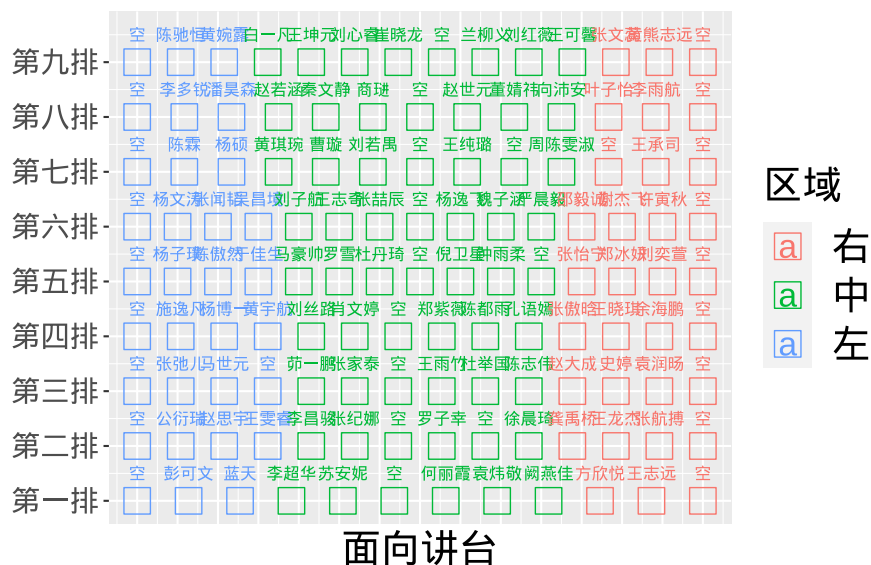
# 设置纵坐标的标注文字
text = c(" 第一排", " 第二排", " 第三排", " 第四排", " 第五排", " 第六排", " 第七排", " 第八排", " 第九排", " 第十排", " 第十一排", " 第十二排", " 第十三排", " 第十四排", " 第十五排")

# 进行最后的数据可视化
ggplot(seat_plot,aes(col,row,color = position)) +
  geom_point(shape = 0,size = 6) +
  # 设置 x 轴和 y 轴的范围以及文字
  scale_x_continuous(limits = c(1,15),breaks = seq(1,15,1)) +
  scale_y_continuous(limits = c(1,9.5),breaks = seq(1,9,1),labels = text) +
  # 设置 x,y 轴的标题, 整个图和图例的标题

```

```
labs(x = " 面向讲台", y = "", color = " 区域", title = " 座位分布图") +
# 增添文字标注, 即增添座位对应的姓名
geom_text(data = seat_name, aes(label = name), size = 3) +
# 对 x, y 轴的元素进行调整
theme(axis.text.y = element_text(size = 15, face = "bold")) +
theme(axis.title.x = element_text(size = 20, face = "bold")) +
theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) +
# 对标题的位置进行调整
theme(plot.title = element_text(size = 25, face = "bold", hjust = 0.5)) +
# 对图例的大小进行调整
theme(legend.key.size = unit(25, 'pt'), legend.text = element_text(size = 20), legend.title = element_text(size = 20)) +
guides(color = guide_legend(override.aes = list(size = 6)))
```

座位分布图



从而得到最终的座位图。

4 最终的输出

尝试将上述功能整合成一个函数，使得输出的结果只由最初的随机数决定，同时在函数中增加功能，根据最初的随机数确定本科生同学在教室中的哪一个位置：

```
# 尝试整合成函数

random_seat = function(seed){
  # 增加判断语句，使得可以根据随机数选择本科生所坐的位置
  undergraduate_position_list = c(" 左"," 中左"," 中右"," 右")
  undergraduta_position = undergraduate_position_list[(seed %% 4)+1]

  # 将之前的代码迁移过来

  # 构造用于抽样数据框
  sample_data_frame = seat_grid
  sample_data_frame = sample_data_frame %>% mutate(name = " 空")
  sample_data_frame = sample_data_frame[!sample_data_frame$labels == " 无",]

  # 设置用于抽样的向量
  n_sample = nrow(sample_data_frame)

  # 根据输入设置随机种子，确保结果可以复现
  set.seed(seed)

  # 抽样更新座位数据框
  sample_order = sample(seq(1:n_sample),size = n_sample,replace = FALSE)
  sample_data_frame = sample_data_frame[sample_order,]

  # 进行迭代排座位
  i = 1
  for(j in 1:130){
```

```
# 设置条件语句，使得本科生按照之前的结果被分配在教室的某一位置
if(sample_data_frame[j,"position"] == undergraduta_position&sample_data_frame[j,"label"] == "空")
    sample_data_frame[j,"name"] = name_list[[i,1]]
    i = i + 1
}
# 安排好本科生之后，跳出此次循环
if(i > 22){
    break
}
}

i = 23
for(j in 1:130){
    if(sample_data_frame[j,"name"] == "空"&sample_data_frame[j,"labels"] == "空"&!sample_data_frame[j,"position"] == "空")
        sample_data_frame[j,"name"] = name_list[[i,1]]
        i = i + 1
    }
    if(i > 92){
        break
    }
}

# 进行数据可视化
seat_plot = data.frame(col = 0 ,
                        row = 0,
                        labels = 0 ,
                        position = 0,
                        name = 0)

# 由于绘图时不需要标注“无”，因此重新放置每个座位的位置
for(i in 1:9){
    # 载入每一排的数据
```

```

seat_row = sample_data_frame[sample_data_frame$row==i,]
# 按照列的大小对每一排的位置进行排序
seat_row = arrange(seat_row,col)
n = nrow(seat_row)
row_name = data.frame(row = c(i),
                      col = seq(1,15,length.out = n),
                      labels = seat_row[,3],
                      position = seat_row[,4],
                      name = seat_row[,5])

# 对每一排的数据进行整合
seat_plot = rbind(seat_plot,row_name)
}

# 对数据框做一些修改，为最终绘图做准备
seat_plot = seat_plot[-1,]
seat_plot[seat_plot$position==" 中左"|seat_plot$position==" 中右",4] = " 中"
seat_plot[seat_plot$labels==" 不坐", "labels"] = " 空"

# 设置用于标注名字的数据框
seat_name = seat_plot
seat_name[,2] = seat_name[,2]+0.5

# 设置坐标轴标注文字
text = c(" 第一排", " 第二排", " 第三排", " 第四排", " 第五排", " 第六排", " 第七排", " 第八排")

# 进行最后的数据可视化
seat_picture = ggplot(seat_plot,aes(col,row,color = position)) +
  geom_point(shape = 0,size = 7) +
  scale_x_continuous(limits = c(1,15),breaks = seq(1,15,1)) +
  scale_y_continuous(limits = c(1,9.5),breaks = seq(1,9,1),labels = text) +
  labs(x = " 面向讲台",y="", color = " 区域",title = " 座位分布图",subtitle = " 同一区域同

```

```

geom_text(data = seat_name,aes(label = name),size = 4.5,fontface = "bold") +
theme(axis.text.y = element_text(size = 15, face = "bold")) +
theme(axis.title.x = element_text(size = 20, face = "bold")) +
theme(axis.ticks.x = element_blank(),axis.text.x = element_blank()) +
theme(plot.title = element_text(size = 25, face = "bold",hjust = 0.5)) +
theme(legend.key.size=unit(25,'pt'),legend.text = element_text(size = 20),legend.titl
guides(color = guide_legend(override.aes = list(size = 6)))

# 设置文件名字，标明所用的随机数
file_name = paste("seat_picture_",seed,".pdf")
ggsave(seat_picture,file = file_name, width = 12, height = 8)

seat_picture
}

```

根据从周围的同学们征集而来的数字，选用 7, 15, 24, 39, 40, 50, 53, 79, 92, 95 十个数字作为最初的随机种子，生成座位，并储存到相应文件夹中。

```

for(i in c(7,15,24,39,40,50,53,79,92,95)){
  random_seat(i)
}

```