# 南京 58 同城招聘网热门岗位数据探索

——数据科学与创新课程设计报告

161271029 岳翔 工程管理学院

## 1. 数据分析任务

薪水的影响因素有哪些呢？为了探究招聘岗位薪水的影响因素，我爬取了南京市 58 同城热门招聘岗位的 2699 条信息，获取了地址、招聘公告名、公司名、薪水、工作岗位类型、学历要求、工作经验要求、福利个数这八类数据，接下来，我对各个因素与薪水的关系进行统计探索，并且运用 K-means 和 DIANA 层次聚类法进行聚类探索，以挖掘数据间的关联和规律。


## 2. 数据探索步骤

### 2.1 数据获取

首先，通过手动翻页可知页面 url 构成由'http://nj.58.com/job/'和“pn(x)”构成，其中 x 为当前页码。因此可以通过循环操作遍历所有页码。为了获取最大页码，分析 HTML 代码可知最大页码是 i.total_page 元素的 HTML 内容。

其次，为了爬取各个相关信息，需要检查 HTML 的代码元素并通过 rvest 来提取 HTML 的 dom 节点。各个信息对应如下:

| 信息名 | 对应 DOM 节点 |
| --- | --- |
| **地址** | span.address |
| **招聘公告名** | span.name |
| **公司名** | div.item_con.job_comp.div.comp_name. a.fl |
| **薪水** | div.item_con.job_comp. p.job_salary |
| **工作岗位类型** | div.item_con.job_comp.span.cate |
| **学历要求** | div.item_con.job_comp.span.xueli |
| **工作经验要求** | div.item_con.job_comp.span.jingyan |
| **福利个数** | length(div.item_con.job_comp.li.job_item.clearfix) |

```
    library(rvest)
    library(mongolite)
    library(stringr)
    library(RJSONIO)
    prefix = 'http://nj.58.com/job/'
    url = paste(prefix, 'pn1', sep='')
    maxPage = url %>% read_html() %>% html_nodes('i.total_page') %>%
html_text() %>% as.numeric()
    address = name = company = salary = type = edubg = experience =
welfareNum = NULL
    # main loop
    for(i in 1:maxPage){
      url = paste(prefix, 'pn', i, sep='')
      html = read_html(url)
      address = c(address, html %>% html_nodes('span.address') %>%
html_text())
      name = c(name, html %>% html_nodes('span.name') %>% html_text())
      requires = html %>% html_nodes('div.item_con.job_comp')
      company = c(company, requires %>% html_nodes('div.comp_name') %>%
html_nodes('a.fl') %>% html_text())
      salary = c(salary, html %>% html_nodes('p.job_salary') %>%
html_text())
      type = c(type, requires %>% html_nodes('span.cate') %>%
html_text())
      edubg = c(edubg, requires %>% html_nodes('span.xueli') %>%
html_text())
      experience = c(experience, requires %>%
html_nodes('span.jingyan') %>% html_text())
      welfareNodes = html %>% html_nodes('li.job_item.clearfix')
      vec = NULL
      for(node in welfareNodes){
        vec = c(vec, node %>% html_nodes('div.job_wel.clearfix') %>%
html_nodes('span') %>% length())
      }
      welfareNum = c(welfareNum, vec)
    }
    df = data.frame(address, name, company, salary, type, edubg,
```

```
experience, welfareNum)
    json_data = toJSON(df)
    write.table(df, '58 同城热门岗位信息.txt', row.names = F, quote = F)
    write.csv(df, '58 同城热门岗位信息.csv')
    conn = mongo(collection = 'jobs', db = 'test', url =
'mongodb://localhost')
    conn$insert(json_data)
    # 注：导出数据库文件是通过 mongodump.exe 执行的
```

如上，通过 R 语言爬虫程序抓取完毕后，转换为 dataframe 格式，再存为 JSON 数据格式，分别保存为 txt 文件和 csv 文件，最后保存至 mongodb 中并导出 BSON 格式的表。

## 2.2 数据清洗和预处理：

**①导入上一次爬取的 csv 数据文件。**

```
df = read.csv("58 同城热门岗位信息.csv", stringsAsFactors = F)
```

**②丢弃没有分析价值的“招聘公告名”、“公司名”、“序号”列。**

```
df$X = df$name = df$company = NULL  #删除序号、标题、公司名
```

**③通过简单分析可见薪水列存在一部分面议值，将这些值置为 NA。**

```
df$salary[df$salary == "面议"] = NA
```

**④因为薪水列的表达是 NA 或者是“xxx-xxx 元/月”，因此用正则表达式来提取薪水，并以前后均值代替原数据。**

```
library(stringr)
meanSalary = rep(0, length(df$salary))
for(i in 1:length(df$salary)){
  str = df$salary[i]
  if(is.na(str)){
    meanSalary[i] = NA
  }
  res = str_match_all(str, "\\d+")
  res = res[[1]]
  meanSalary[i] = c(as.numeric(res[1]), as.numeric(res[2])) %>%
mean()
}
df = data.frame(df, meanSalary)
```

**⑤对薪水缺失值进行插值。使用 mice 包。**

```
library(mice)
library(VIM)
dfUninterp = df
df$salary = NULL
df = df %>% mice(seed = 999) %>% complete(action = 5)
df$meanSalary.1 = NULL
```

## 2.3 统计薪水平均数 top50 特征

**绘制各个特征 top 50 平均薪水图。由于各个特征具有很强的离散性质，因此可以先简单统计出各项特征的薪水值 top50 的情况。**

```
    library(ggplot2)
    #绘制各个特征top 50 平均薪水图
    for(item in c("address", "type", "edubg", "experience",
"welfareNum")){
      meanSalaryOfEach = tapply(df$meanSalary, df[item], mean)
      meanSalaryOfEach = data.frame(rownames(meanSalaryOfEach),
meanSalaryOfEach)
      rownames(meanSalaryOfEach) =
1:length(meanSalaryOfEach$rownames.meanSalaryOfEach.)
      names(meanSalaryOfEach) = c(item, "meanSalary")
      elder = meanSalaryOfEach
      meanSalaryOfEach = meanSalaryOfEach[order(elder[2], decreasing =
T),]
      base1 = switch (item,
        "address" = ggplot(meanSalaryOfEach[1:50,],
aes(x=reorder(address, meanSalary), meanSalary)),
        "type" = ggplot(meanSalaryOfEach[1:50,], aes(x=reorder(type,
meanSalary), meanSalary)),
        "edubg" = ggplot(meanSalaryOfEach[1:50,], aes(x=reorder(edubg,
meanSalary), meanSalary)),
        "experience" = ggplot(meanSalaryOfEach[1:50,],
aes(x=reorder(experience, meanSalary), meanSalary)),
        "welfareNum" = ggplot(meanSalaryOfEach[1:50,],
aes(x=reorder(welfareNum, meanSalary), meanSalary))
      )
      plot1 = base1 + geom_bar(stat = "identity", aes(fill=meanSalary))
+ theme(axis.text.x = element_text(angle = 90))
      plot1 = plot1 + labs(x=item, title=paste(item,"vs mean salary"))
      plot1 = plot1 + scale_size_area()
      print(plot1)
      meanSalaryOfEach = merge(meanSalaryOfEach, df, all.x = T, by =
item)
      base2 = switch (item,
        "address" = ggplot(df,aes(address, meanSalary)),
        "type" = ggplot(df, aes(type, meanSalary)),
        "edubg" = ggplot(df, aes(edubg, meanSalary)),
        "experience" = ggplot(df, aes(experience, meanSalary)),
        "welfareNum" = ggplot(df, aes(welfareNum, meanSalary))
      )
      print(base2 + geom_boxplot())
    }
```

经过上述处理得到的图像将在结论小节中进行分析。

## 2.4 相关分析

**由于数据离散程度较大，为了实现相关分析，必须转换成数值类型。因此，我考虑用频数和自定义的水平代替原文本。最后，通过 corrplot 绘制相关系数矩阵。**

```
    #学历要求 -> 教育等级
    for(i in 1:length(df$edubg)){
      str = df$edubg[i]
      eduLevel[i] = switch (str,
        "不限" = 0,
```

```
    "技校" = 3,
    "中专" = 6,
    "高中" = 9,
    "大专" = 12,
    "本科" = 15
  )
}
#经验 -> 经验年
for(i in 1:length(df$edubg)){
  str = df$experience[i]
  if(str %in% c('不限', '10 年以上', '1 年以下')){
    meanExp[i] = switch (str,
      '10 年以上' = 10,
      '1 年以下' = 0.5,
      '不限' = 0
    )
    next
  }
  res = str_match_all(str, "\\d+")
  res = res[[1]]
  meanExp[i] = c(as.numeric(res[1]), as.numeric(res[2])) %>% mean()
}
df = data.frame(df, meanExp, meanSalary, eduLevel) #合成新数据框
df$experience = df$salary = df$edubg = NULL #丢弃原列
#地址 -> 所在地公司数量
addressFreq = count(df$address)
names(addressFreq) = c("address", "addressFreq")
df = merge(df, addressFreq, by.x = "address", by.y = "address")
df$address = NULL #删除 address
#工作类型 -> 工作频数
typeFreq = count(df$type)
names(typeFreq) = c("type", "typeFreq")
df = merge(df, typeFreq, by.x = "type", by.y = "type")
df$meanSalary.1 = df$type = NULL #删除 type
#打印相关关系图
corrplot(cor(df),method="shade",addCoef.col="black",order="AOE")
```

## 2.5 招聘岗位聚类

**不同薪水区间的岗位存在一定的内部相似度，因此在这里通过 K-means 聚类方法，将岗位按照 1~15 簇进行聚类，计算内相似度，找出聚类最好的簇个数。并且将薪水和各个因素两两进行可视化绘制。**

```
#K-means 对招聘信息品质进行分类
wss = numeric(15)
for (k in 1:15)
  wss[k] <- sum(kmeans(df, centers=k, nstart=25)$withinss)
plot(1:15, wss, type="o", xlab="Number of Clusters", ylab="Within
Sum of Squares")
#k = 4 时聚类效果最佳
kMeansEquals4 = kmeans(df, centers=4, nstart=25)
#依次绘制二维聚类可视化图表
df$cluster = factor(kMeansEquals4$cluster)
```

```
    centers=as.data.frame(kMeansEquals4$centers)
    g1 = ggplot(data=df, aes(x=meanSalary, y=addressFreq,
color=cluster)) +
      geom_point() + theme(legend.position="right") +
      geom_point(data=centers, aes(x=meanSalary,y=addressFreq,
color=as.factor(c(1,2,3,4))),
                 size=10, alpha=0.3, show.legend=FALSE)
    g2 = ggplot(data=df, aes(x=meanSalary, y=eduLevel, color=cluster)) +
      geom_point() + theme(legend.position="right") +
      geom_point(data=centers, aes(x=meanSalary,y=eduLevel,
color=as.factor(c(1,2,3,4))),
                 size=10, alpha=0.3, show.legend=FALSE)
    g3 = ggplot(data=df, aes(x=meanSalary, y=welfareNum, color=cluster))
+
      geom_point() + theme(legend.position="right") +
      geom_point(data=centers, aes(x=meanSalary,y=welfareNum,
color=as.factor(c(1,2,3,4))),
                 size=10, alpha=0.3, show.legend=FALSE)
    g4 = ggplot(data=df, aes(x=meanSalary, y=meanExp, color=cluster)) +
      geom_point() + theme(legend.position="right") +
      geom_point(data=centers, aes(x=meanSalary,y=meanExp,
color=as.factor(c(1,2,3,4))),
                 size=10, alpha=0.3, show.legend=FALSE)
    g5 = ggplot(data=df, aes(x=meanSalary, y=typeFreq, color=cluster)) +
      geom_point() + theme(legend.position="right") +
      geom_point(data=centers, aes(x=meanSalary,y=typeFreq,
color=as.factor(c(1,2,3,4))),
                 size=10, alpha=0.3, show.legend=FALSE)
    print(g1)
    print(g2)
    print(g3)
    print(g4)
    print(g5)
```

### 2.6 薪水 top25 招聘岗位层次聚类

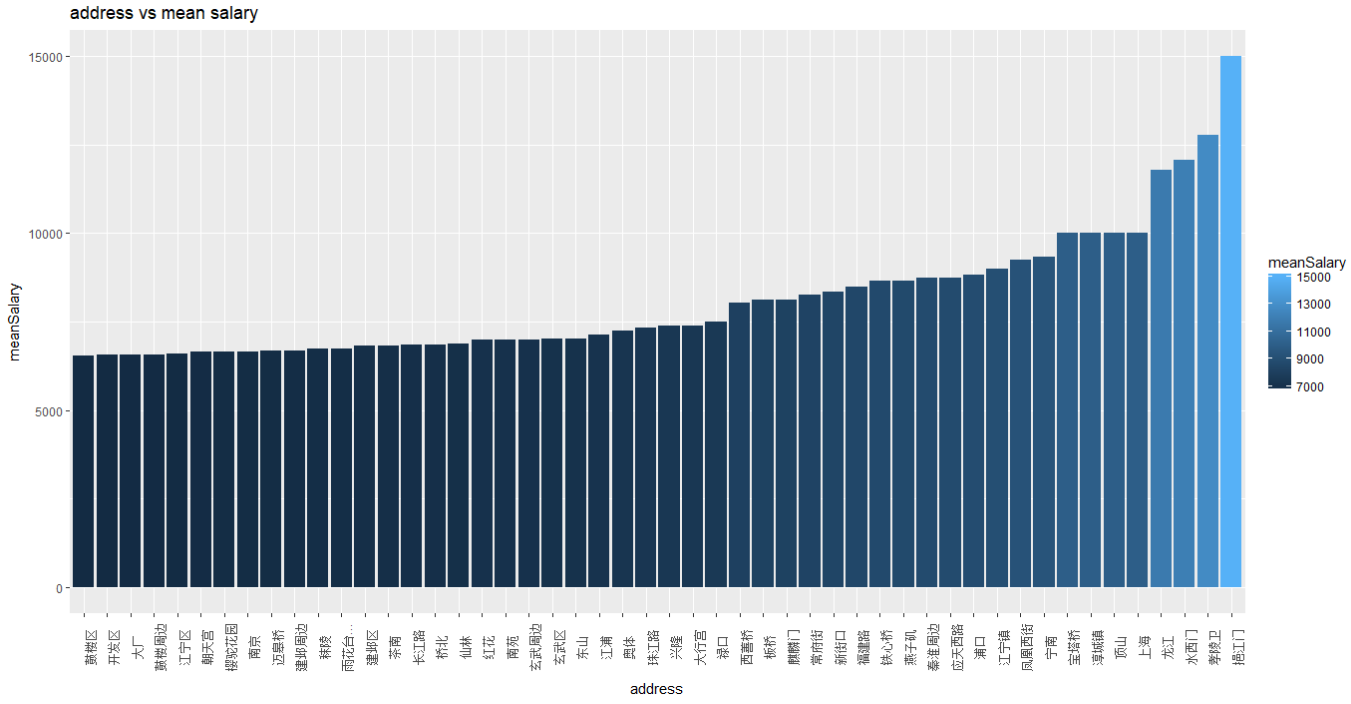
**最后，我们将薪水位于前 25 的岗位信息进行层次聚类。**

```
    #层次聚类
    df$name = theName
    df$meanSalary = as.numeric(df$meanSalary)
    df = df[order(df$meanSalary, decreasing=T), ]
    df = df[1:25, ]
    rownames(df) = df$name
    hc = hclust(dist(df))
    #plot(hc, hang = -1)
    #hcd = as.dendrogram(hc)
    plot(hc)
```

## 3. 结论

**① 平均薪水 top50 的公司所在地分布**

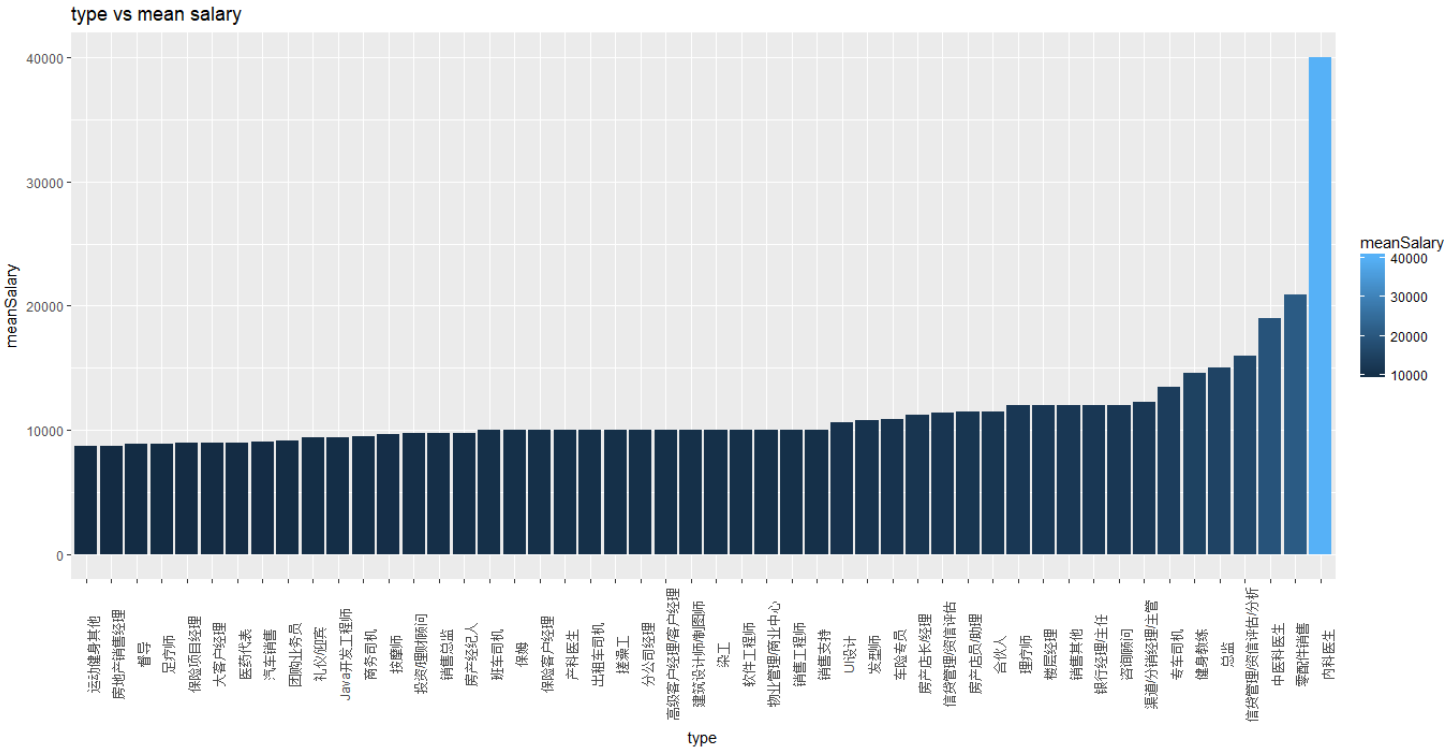
由图中可以看到坐落在挹江门、孝陵卫、水西门的企业平均薪水较高。就业者可以考虑关注这些地方的企业。但是，挹江门可能是由于某些少数薪水较高的职位导致的薪水高，不具有普遍性。




### ② 平均薪水 top50 的职业

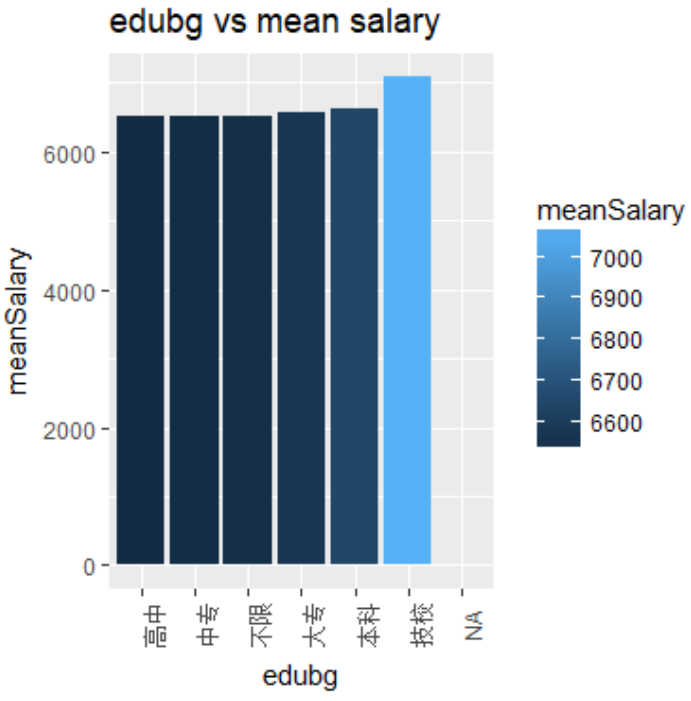
由图可以得知内科医生薪水最高，零配件销售、中医科医生、借贷管理评估这些对专业要求较高的职业平均薪水分布靠前。大多职业的均薪处在 10000 元以下。有 20 个职业均薪位于 10000 元以上。

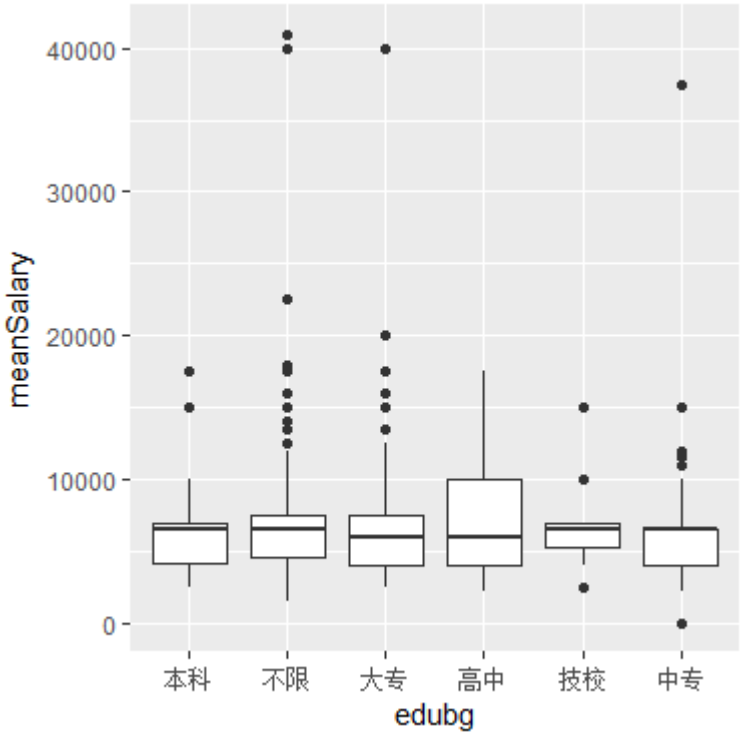
在这里我们可以发现内科医生和前一张图“平均薪水 top50 的公司所在地分布”的最高均薪所在地薪水一致，这就验证了岗位信息中挹江门恰好是内科医生这个岗位，不具有代表性。




### ③ 学历背景均薪统计

在条形统计图中，我们看到技校学历的收入均薪最高，其他的学历均薪基本



持平。这可能是因为平均数无法代表普遍情况的缘故。该图无法反映出学历与均薪的情况。

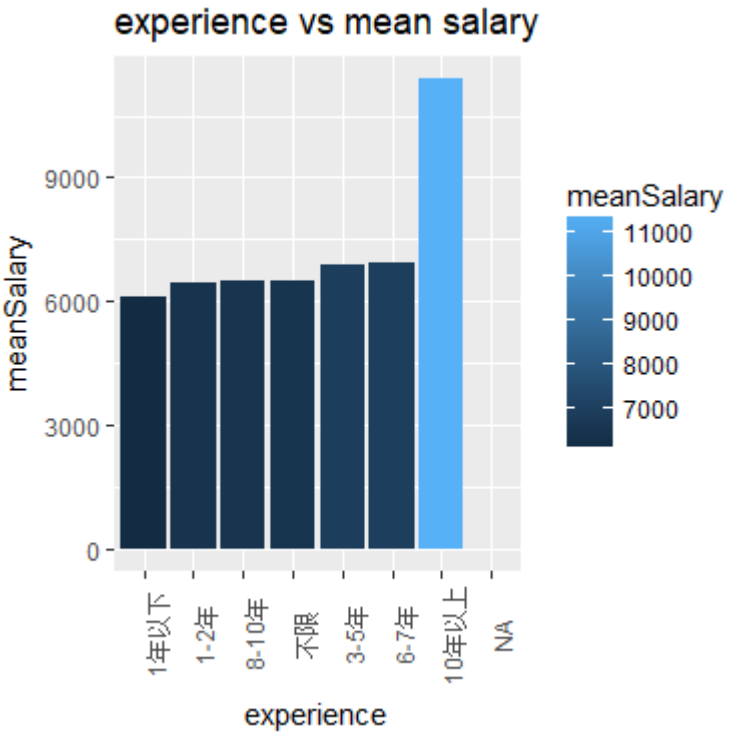
在箱型图中，我们可以看到各个学历的中位数薪水基本持平。各个学历都有少数高收入职位。这可以初步说明学历并不能代表一切，薪水的多少也和个人能力和职位选择挂钩。总体而言，我们可以看到脱颖而出的高收入者都被标记为了异常点，也就是说高收入职位本身就是较少的。

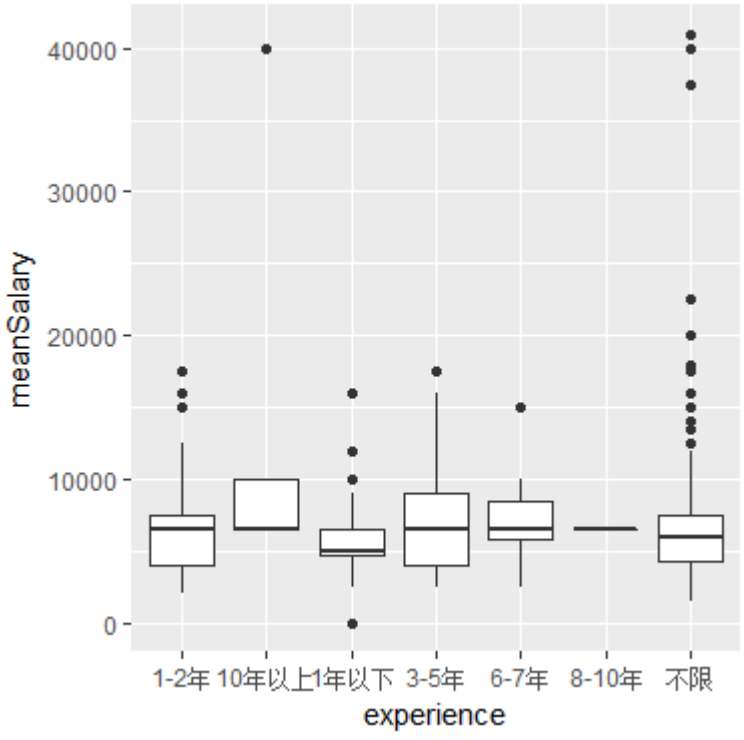
**④ 工作经验均薪统计**

在统计图中我们可以看到工作经验要求在“10 年以上”的职位均薪较高，这是合乎常识的，而其他工作经验的均薪分布在 6000-6500 左右。从该图中我们能得到的信息较少。

在箱型图中，首先我们可以发现，均薪很高的“10 年以上”，是由于一个均薪非常高的异常点所导致的，初步推断是“内科医生”职位，再加上样本较少的因素，导致均值偏高。观察其他工作经验，我们也能发现如下结论：“不限”中，有较少薪水较高的脱颖而出的职位，但是由于样本基数大，均值较为普通，个人认为这是因为招聘信息填写“不限”的职位较多导致的。“8-10 年”样本较少，分布也密集。而七年工作经验以下的样本，没有超过 20000 元的职位，只有少数超过 10000 元的职位。

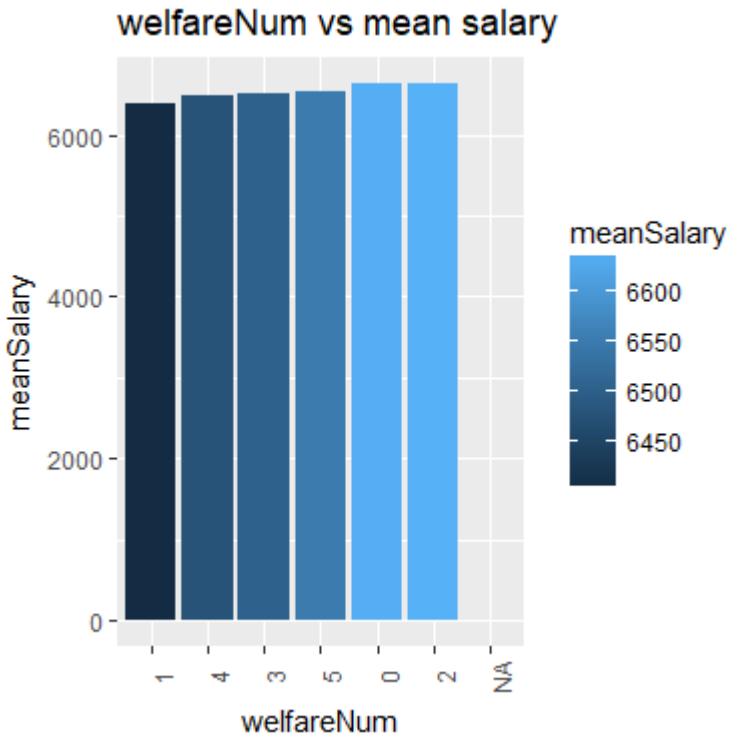
总而言之，高薪职位永远是少数，工作经验各行各业要求不同，不能以统计的方式得出有效结论。



### ⑤ 福利个数均薪统计

从如下条形图中，我们依然无法得到太多有效信息。福利数量可能和薪水关系不大。




### ⑥ 相关程度分析

相关系数矩阵作图如下：

因为数据的高度离散化，我们需要将数据进行数值化处理才能继续探索。在这里，我简单地将各个特征的频数代替原特征，虽然数值化了，但是仍然不是连续的。特征的高度离散化特性依然存在，这可能会导致相关分析的失败。

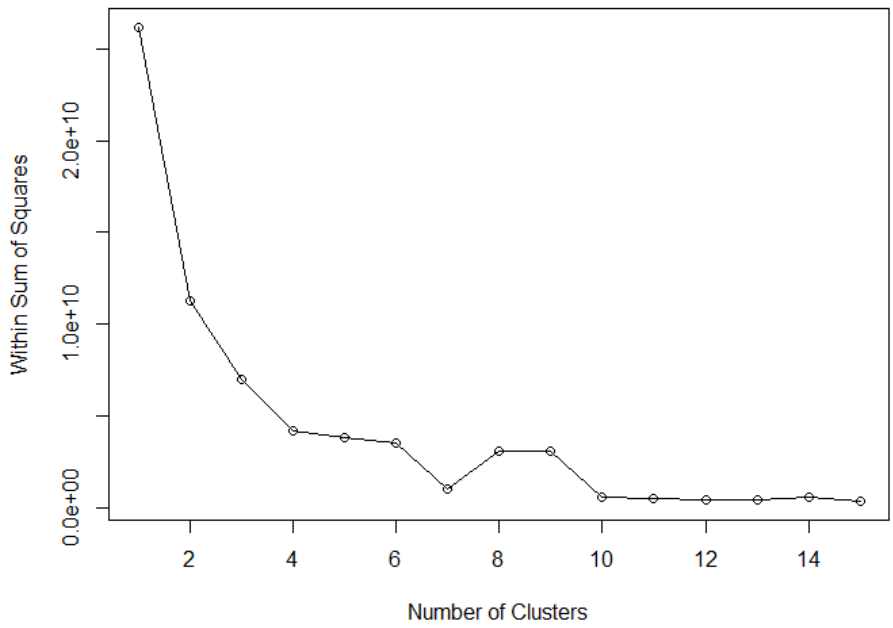
作图后，我们发现 meanSalary 和其他特征相关系数接近 0，这说明基本是没有关系的。

得到这个结论有两点因素：首先，招聘信息大多以离散型特征或指示型特征存在，薪水与他们之间不存在一一对应的关系。其次，我的数值化处理方法是用频数代替特征文本，原先文本就和薪水没有直接的相关关系，而得到的频数自然也没有相关关系，因此最终相关系数接近 0。

如果想分析热门岗位薪水的影响因素，应进一步对特征进行分解，或者找到与薪水波动相关的数值型变量。但是由于 58 同城提供的信息有限，除了薪水，并没有给出数值型的招聘信息，因此可供分析的东西有限。

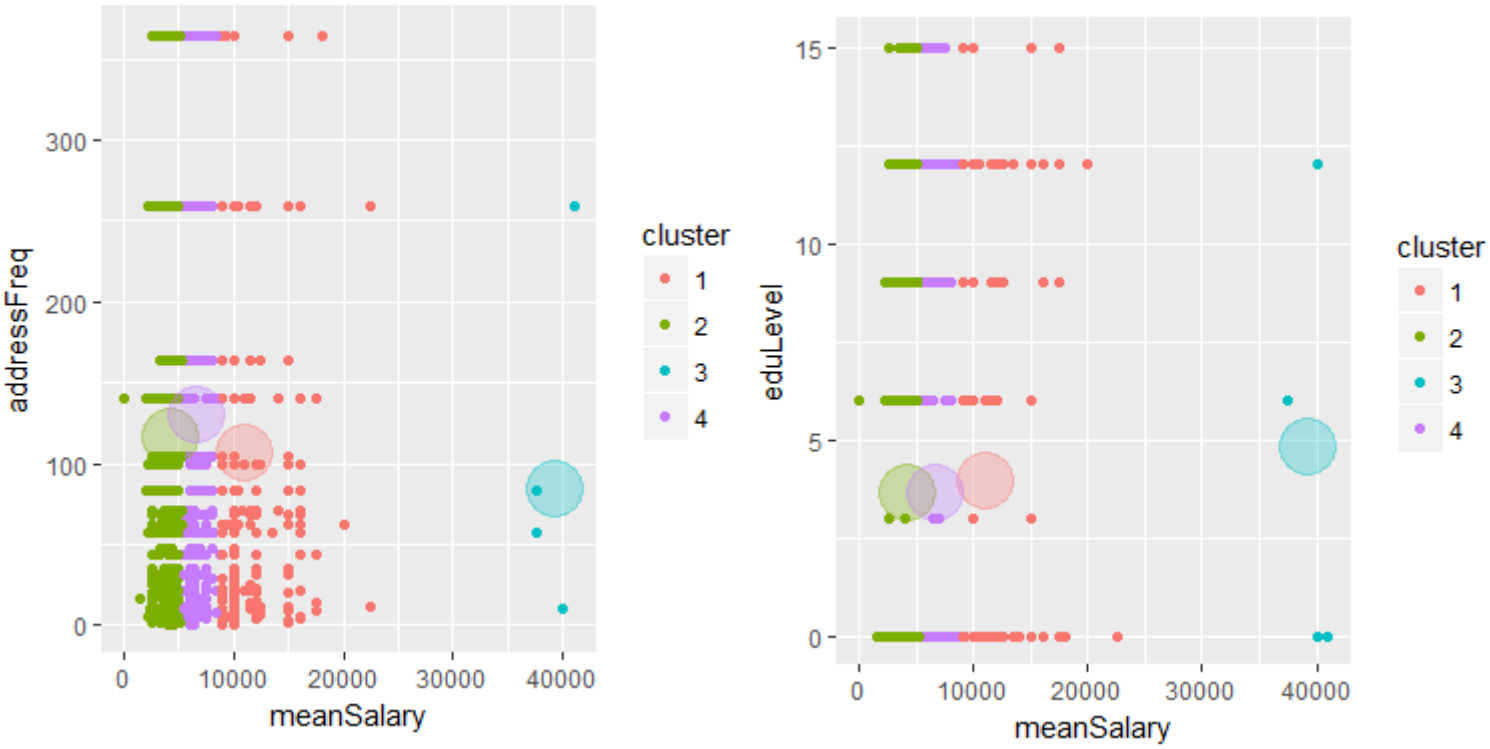
**⑦ K-means 聚类分析**

首先，我们通过尝试 1~15 簇聚类，将内相似度绘制出来。结果如下图：

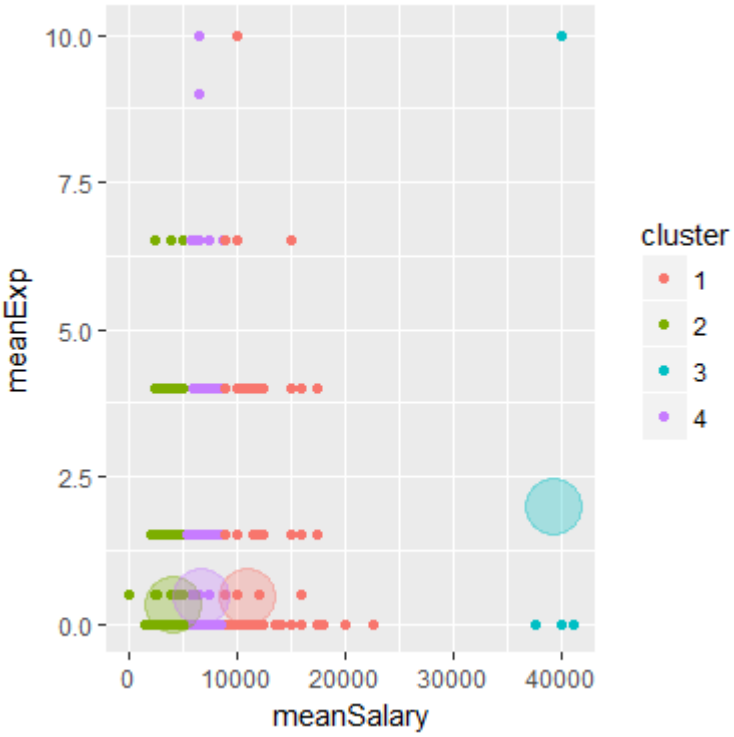



可以看到 k = 4 时，聚类效果最佳。也就是说，我们可以假定目前招聘岗位被分为了四个不同的水平：优、良、普通、差。

接下来我们绘制可视化聚类关系图。x 统一为 meanSalary，而 y 为其他因素。



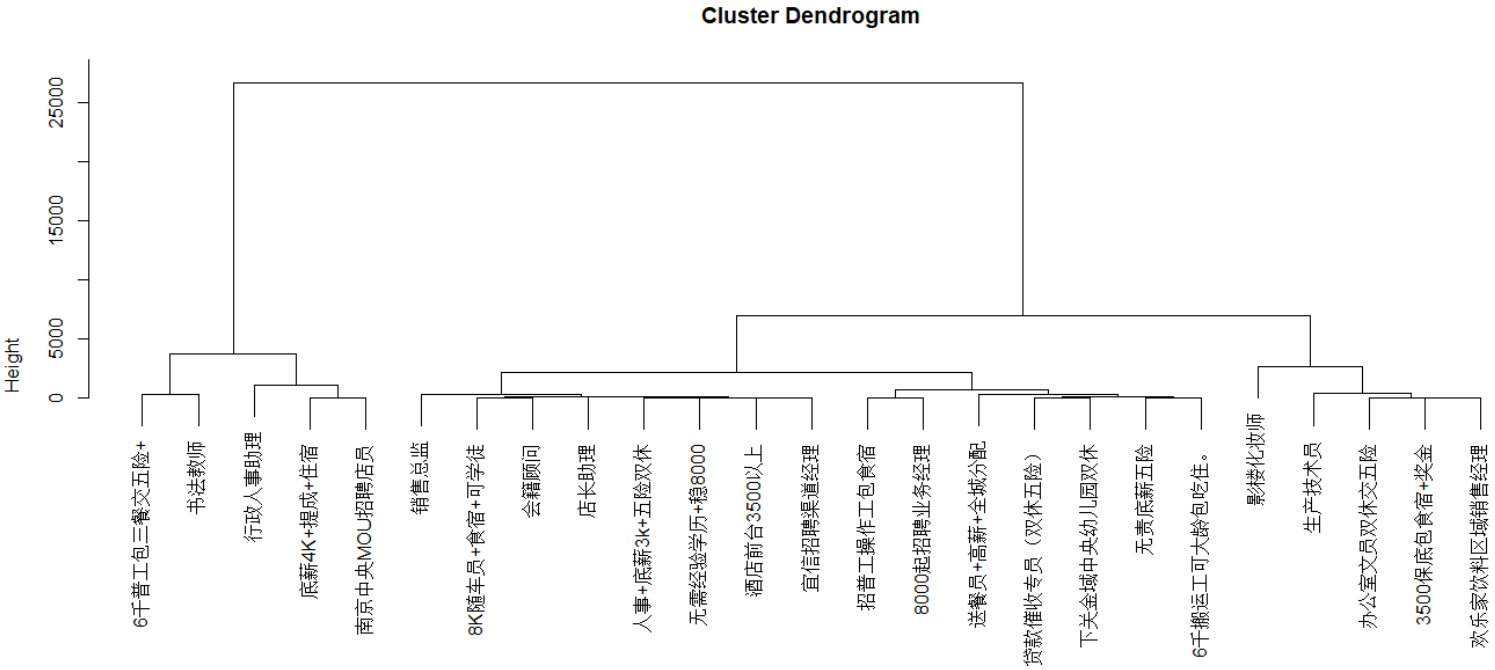

通过以上聚类相关因素图，我们可以得到以下几个结论：首先，聚类数据基



本划分正确，主要依靠薪水的四个大体区间进行了划分。其次，薪水越低，同水平岗位也就越多。这说明岗位越优质数量则越少。最后，可以大体看出，薪水是划分岗位质量的主要标准，其他的因素对于岗位质量基本没有太大的影响。

### ⑧ 层次聚类分析




由于样本个数过多，在这里仅选取了薪水数 top25 的岗位进行聚类。可以看到岗位之间存在着一定的内部联系。