共三组数据，其中jobinfo.xlsx需要预先做一些清理与预处理，另外两组无需。如下具体的命令供大家参考使用：

1) windows系统：

rm(list = ls()) # 清空工作空间

### 从jobinfo.xlsx 到 数据分析岗位招聘.csv ###

# install.packages(readxl)

library(readxl)

# install.packages(ggplot2)

library(ggplot2)

# install.packages(jiebaR)

library(jiebaR)

library(readxl)

library(ggplot2)

install.packages(jiebaR)

library(jiebaR)

jobinfo = read\_excel("D:\\document\\mydocument\\R\\jobinfo\\jobinfo.xlsx",sheet=1) # 读取原始数据,把这个路径修改了可以直接拷贝到R运行

str(jobinfo) # 查看数据结构

## Classes 'tbl\_df', 'tbl' and 'data.frame': 7370 obs. of 12 variables:

## $ 职位 : chr "数据分析师" "商品主管/货品管理/数据分析/买手" "研究员（数据分析 海归优先）" "口碑-大区中台数据分析 (职位编号：GP030158)" ...

## $ 公司名称: chr "北京中技所知识产权服务有限公司" "华歌尔（中国）时装有限公司" "央视市场研究股份有限公司" "阿里巴巴集团" ...

## $ 地区 : chr "北京" "北京" "北京" "北京" ...

## $ 公司类别: chr "国企" "外资" "合资" "上市公司" ...

## $ 公司规模: chr "50-150人" "1000-5000人" "500-1000人" "10000人以上" ...

## $ 行业类别: chr "金融/投资/证券" "服装/纺织/皮革" "专业服务(咨询、人力资源、财会)" "互联网/电子商务" ...

## $ 经验 : num 0 0 2 3 2 0 2 1 1 1 ...

## $ 学历 : chr "本科" "大专" "本科" "本科" ...

## $ 人数 : chr "招聘若干人" "招聘若干人" "招聘1人" "招聘1人" ...

## $ 描述 : chr "岗位说明 运营分析 输出业务分析报告 对业务问题进行深入分析 为公司运营决策 产品方向 销售策略提供数据支持 全业务线"| \_\_truncated\_\_ "职位描述 负责配置单店基本库存 负责全公司的货品调配 产品数据分析职位要求 至 岁之间 大专及以上学历 至少 年以上服"| \_\_truncated\_\_ "岗位职责 工作职责 为了协助研究经理完成业绩目标 根据部门工作量和工作任务 参与项目研究 承担项目调查 分析数据 撰写"| \_\_truncated\_\_ "建立良好的产品数据监测和汇报机制 对产品数据进行例行监控 及时发现数据的异常波动 并通过深入分析了解其原因 深入理"| \_\_truncated\_\_ ...

## $ 最低薪资: chr "8000" "NA" "NA" "NA" ...

## $ 最高薪资: chr "NA" "NA" "NA" "NA" ...

## (1) 构造因变量：平均薪资的变量 ##

jobinfo$最低薪资 = as.numeric(jobinfo$最低薪资) # 将最低薪资的字符型变量改为数值型变量

jobinfo$最高薪资 = as.numeric(jobinfo$最高薪资) # 将最高薪资的字符型变量改为数值型变量

# 在jobinfo中创建平均薪资的变量

jobinfo$平均薪资 = (jobinfo$最高薪资 + jobinfo$最低薪资) / 2

## (2) 按照disctrict向量将地区重新划分为北上深和非北上深两个水平 ##

loc = which(jobinfo$地区 %in% c("北京", "上海", "深圳"))

loc\_other = which(!jobinfo$地区 %in% c("北京", "上海", "深圳"))

jobinfo$地区[loc] = 1

jobinfo$地区[loc\_other] = 0

jobinfo$地区 = as.numeric(jobinfo$地区)

## (3) 将公司规模转化为因子型变量，便于画图 ##

jobinfo$公司规模 = factor(jobinfo$公司规模, levels = c("少于50人", "50-150人", "150-500人", "500-1000人", "1000-5000人", "5000-10000人", "10000人以上"))

levels(jobinfo$公司规模)[c(2, 3)] = c("50-500人", "50-500人")

# 将50-150人和150-500人合并为一个水平：50-500人

## (4) 将学历转化为因子型变量，便于画图 ##

jobinfo$学历 = factor(jobinfo$学历, levels = c("中专", "高中", "大专", "无", "本科", "硕士", "博士"))

## (5) 匹配各个公司要求的统计软件 ##

# 首先建立software数据框，用于存放各个公司的软件匹配结果

software = as.data.frame(matrix(0, nrow = length(jobinfo$描述), ncol = 12)) # 先建立一个0矩阵，行数为观测数，列数为统计软件的个数，并转化为data frame格式

colnames(software) = c("R", "SPSS", "Excel", "Python", "MATLAB", "Java", "SQL", "SAS", "Stata", "EViews", "Spark", "Hadoop") # 将software的data frame的列名改为软件名称

mixseg = worker() # 按照缺省值，设置分词引擎

# 对每个描述观测进行分词，并存储在software里面，循环次数为总观测数，总观测数可通过length(jobinfo$描述)获取

for (j in 1:length(jobinfo$描述)){

subdata = as.character(jobinfo$描述[j]) # 取出每个观测，保存在subdata变量

fenci = mixseg[subdata] # 对取出的观测进行分词，保存在分词变量

# 设置各个软件的判别条件，以R为例，R.indentify表示r或R是否在fenci这个变量里

R.identify = ("R" %in% fenci) | ("r" %in% fenci)

SPSS.identify = ("spss" %in% fenci) | ("Spss" %in% fenci) | ("SPSS" %in% fenci)

Excel.identify = ("excel" %in% fenci) | ("EXCEL" %in% fenci) | ("Excel" %in% fenci)

Python.identify = ("Python" %in% fenci) | ("python" %in% fenci) | ("PYTHON" %in% fenci)

MATLAB.identify = ("matlab" %in% fenci) | ("Matlab" %in% fenci) | ("MATLAB" %in% fenci)

Java.identify = ("java" %in% fenci) | ("JAVA" %in% fenci) | ("Java" %in% fenci)

SQL.identify = ("SQL" %in% fenci) | ("Sql" %in% fenci) | ("sql" %in% fenci)

SAS.identify = ("SAS" %in% fenci) | ("Sas" %in% fenci) | ("sas" %in% fenci)

Stata.identify = ("STATA" %in% fenci) | ("Stata" %in% fenci) | ("stata" %in% fenci)

EViews.identify = ("EViews" %in% fenci) | ("EVIEWS" %in% fenci) | ("Eviews" %in% fenci) | ("eviews" %in% fenci)

Spark.identify = ("Spark" %in% fenci) | ("SPARK" %in% fenci) | ("spark" %in% fenci)

Hadoop.identify = ("HADOOP" %in% fenci) | ("Hadoop" %in% fenci) | ("hadoop" %in% fenci)

# 判断各个描述变量里面是否有某软件要求，以R为例，第j个描述变量，若R.identify为TRUE时，software的第j行的R变量为1，反之为0；

# 1表示有要求，0表示无要求

if (R.identify) software$R[j] = 1

if (SPSS.identify) software$SPSS[j] = 1

if (Excel.identify) software$Excel[j] = 1

if (Python.identify) software$Python[j] = 1

if (MATLAB.identify) software$MATLAB[j] = 1

if (Java.identify) software$Java[j] = 1

if (SQL.identify) software$SQL[j] = 1

if (SAS.identify) software$SAS[j] = 1

if (Stata.identify) software$Stata[j] = 1

if (EViews.identify) software$EViews[j] = 1

if (Spark.identify) software$Spark[j] = 1

if (Hadoop.identify) software$Hadoop[j] = 1

}

# 将平均薪资和software这两个数据框合并

jobinfo.new = cbind(jobinfo$平均薪资, software)

colnames(jobinfo.new) = c("平均薪资", colnames(software))

## (6) 加入需要的变量 ##

# 地区

jobinfo.new$地区 = jobinfo$地区

# 公司类别

jobinfo.new$公司类别 = jobinfo$公司类别

# 公司规模

jobinfo.new$公司规模 = jobinfo$公司规模

# 学历

jobinfo.new$学历 = jobinfo$学历

# 要求经验

jobinfo.new$经验要求 = jobinfo$经验

# 行业类别

jobinfo.new$行业类别 = jobinfo$行业类别

## (7) 处理观测：公司类别中，非营利机构与事业单位两子类观测过少，没有对比价值，予以删除 ##

table(jobinfo.new$公司类别)

##

## 创业公司 非营利机构 国企 合资 民营公司 上市公司

## 90 20 291 762 4917 406

## 事业单位 外资

## 15 869

jobinfo.new = jobinfo.new[-which(jobinfo.new$公司类别 %in% c("非营利机构", "事业单位")), ]

## (8) 重赋列名 ##

colnames(jobinfo.new) = c("aveSalary", colnames(jobinfo.new[2:13]), "area", "compVar", "compScale", "academic", "exp", "induCate")

## (9) 保存做过预处理的数据集 ##

write.csv(jobinfo.new, file = "数据分析岗位招聘.csv", row.names = FALSE)

2) mac系统：

rm(list = ls()) # 清空工作空间

### 数据预处理：从jobinfo.xlsx 到 数据分析岗位招聘.csv ###

# install.packages(readxl)

library(readxl)

# install.packages(ggplot2)

library(ggplot2)

# install.packages(jiebaR)

library(jiebaR)

jobinfo = read\_excel("jobinfo.xlsx") # 读取原始数据

str(jobinfo) # 查看数据结构

## Classes 'tbl\_df', 'tbl' and 'data.frame': 7370 obs. of 12 variables:

## $ 职位 : chr "数据分析师" "商品主管/货品管理/数据分析/买手" "研究员（数据分析 海归优先）" "口碑-大区中台数据分析 (职位编号：GP030158)" ...

## $ 公司名称: chr "北京中技所知识产权服务有限公司" "华歌尔（中国）时装有限公司" "央视市场研究股份有限公司" "阿里巴巴集团" ...

## $ 地区 : chr "北京" "北京" "北京" "北京" ...

## $ 公司类别: chr "国企" "外资" "合资" "上市公司" ...

## $ 公司规模: chr "50-150人" "1000-5000人" "500-1000人" "10000人以上" ...

## $ 行业类别: chr "金融/投资/证券" "服装/纺织/皮革" "专业服务(咨询、人力资源、财会)" "互联网/电子商务" ...

## $ 经验 : num 0 0 2 3 2 0 2 1 1 1 ...

## $ 学历 : chr "本科" "大专" "本科" "本科" ...

## $ 人数 : chr "招聘若干人" "招聘若干人" "招聘1人" "招聘1人" ...

## $ 描述 : chr "岗位说明 运营分析 输出业务分析报告 对业务问题进行深入分析 为公司运营决策 产品方向 销售策略提供数据支持 全业务线"| \_\_truncated\_\_ "职位描述 负责配置单店基本库存 负责全公司的货品调配 产品数据分析职位要求 至 岁之间 大专及以上学历 至少 年以上服"| \_\_truncated\_\_ "岗位职责 工作职责 为了协助研究经理完成业绩目标 根据部门工作量和工作任务 参与项目研究 承担项目调查 分析数据 撰写"| \_\_truncated\_\_ "建立良好的产品数据监测和汇报机制 对产品数据进行例行监控 及时发现数据的异常波动 并通过深入分析了解其原因 深入理"| \_\_truncated\_\_ ...

## $ 最低薪资: chr "8000" "NA" "NA" "NA" ...

## $ 最高薪资: chr "NA" "NA" "NA" "NA" ...

## (1) 构造因变量：平均薪资的变量 ##

jobinfo$最低薪资 = as.numeric(jobinfo$最低薪资) # 将最低薪资的字符型变量改为数值型变量

jobinfo$最高薪资 = as.numeric(jobinfo$最高薪资) # 将最高薪资的字符型变量改为数值型变量

# 在jobinfo中创建平均薪资的变量

jobinfo$平均薪资 = (jobinfo$最高薪资 + jobinfo$最低薪资) / 2

## (2) 按照disctrict向量将地区重新划分为北上深和非北上深两个水平 ##

loc = which(jobinfo$地区 %in% c("北京", "上海", "深圳"))

loc\_other = which(!jobinfo$地区 %in% c("北京", "上海", "深圳"))

jobinfo$地区[loc] = 1

jobinfo$地区[loc\_other] = 0

jobinfo$地区 = as.numeric(jobinfo$地区)

## (3) 将公司规模转化为因子型变量，便于画图 ##

jobinfo$公司规模 = factor(jobinfo$公司规模, levels = c("少于50人", "50-150人", "150-500人", "500-1000人", "1000-5000人", "5000-10000人", "10000人以上"))

levels(jobinfo$公司规模)[c(2, 3)] = c("50-500人", "50-500人")

# 将50-150人和150-500人合并为一个水平：50-500人

#par(family = 'STKaiti')

## (4) 将学历转化为因子型变量，便于画图 ##

jobinfo$学历 = factor(jobinfo$学历, levels = c("中专", "高中", "大专", "无", "本科", "硕士", "博士"))

## (5) 匹配各个公司要求的统计软件 ##

# 首先建立software数据框，用于存放各个公司的软件匹配结果

software = as.data.frame(matrix(0, nrow = length(jobinfo$描述), ncol = 12)) # 先建立一个0矩阵，行数为观测数，列数为统计软件的个数，并转化为data frame格式

colnames(software) = c("R", "SPSS", "Excel", "Python", "MATLAB", "Java", "SQL", "SAS", "Stata", "EViews", "Spark", "Hadoop") # 将software的data frame的列名改为软件名称

mixseg = worker() # 按照缺省值，设置分词引擎

# 对每个描述观测进行分词，并存储在software里面，循环次数为总观测数，总观测数可通过length(jobinfo$描述)获取

for (j in 1:length(jobinfo$描述)){

subdata = as.character(jobinfo$描述[j]) # 取出每个观测，保存在subdata变量

fenci = mixseg[subdata] # 对取出的观测进行分词，保存在分词变量

# 设置各个软件的判别条件，以R为例，R.indentify表示r或R是否在fenci这个变量里

R.identify = ("R" %in% fenci) | ("r" %in% fenci)

SPSS.identify = ("spss" %in% fenci) | ("Spss" %in% fenci) | ("SPSS" %in% fenci)

Excel.identify = ("excel" %in% fenci) | ("EXCEL" %in% fenci) | ("Excel" %in% fenci)

Python.identify = ("Python" %in% fenci) | ("python" %in% fenci) | ("PYTHON" %in% fenci)

MATLAB.identify = ("matlab" %in% fenci) | ("Matlab" %in% fenci) | ("MATLAB" %in% fenci)

Java.identify = ("java" %in% fenci) | ("JAVA" %in% fenci) | ("Java" %in% fenci)

SQL.identify = ("SQL" %in% fenci) | ("Sql" %in% fenci) | ("sql" %in% fenci)

SAS.identify = ("SAS" %in% fenci) | ("Sas" %in% fenci) | ("sas" %in% fenci)

Stata.identify = ("STATA" %in% fenci) | ("Stata" %in% fenci) | ("stata" %in% fenci)

EViews.identify = ("EViews" %in% fenci) | ("EVIEWS" %in% fenci) | ("Eviews" %in% fenci) | ("eviews" %in% fenci)

Spark.identify = ("Spark" %in% fenci) | ("SPARK" %in% fenci) | ("spark" %in% fenci)

Hadoop.identify = ("HADOOP" %in% fenci) | ("Hadoop" %in% fenci) | ("hadoop" %in% fenci)

# 判断各个描述变量里面是否有某软件要求，以R为例，第j个描述变量，若R.identify为TRUE时，software的第j行的R变量为1，反之为0；

# 1表示有要求，0表示无要求

if (R.identify) software$R[j] = 1

if (SPSS.identify) software$SPSS[j] = 1

if (Excel.identify) software$Excel[j] = 1

if (Python.identify) software$Python[j] = 1

if (MATLAB.identify) software$MATLAB[j] = 1

if (Java.identify) software$Java[j] = 1

if (SQL.identify) software$SQL[j] = 1

if (SAS.identify) software$SAS[j] = 1

if (Stata.identify) software$Stata[j] = 1

if (EViews.identify) software$EViews[j] = 1

if (Spark.identify) software$Spark[j] = 1

if (Hadoop.identify) software$Hadoop[j] = 1

}

# 将平均薪资和software这两个数据框合并

jobinfo.new = cbind(jobinfo$平均薪资, software)

colnames(jobinfo.new) = c("平均薪资", colnames(software))

## (6) 加入需要的变量 ##

# 地区

jobinfo.new$地区 = jobinfo$地区

# 公司类别

jobinfo.new$公司类别 = jobinfo$公司类别

# 公司规模

jobinfo.new$公司规模 = jobinfo$公司规模

# 学历

jobinfo.new$学历 = jobinfo$学历

# 要求经验

jobinfo.new$经验要求 = jobinfo$经验

# 行业类别

jobinfo.new$行业类别 = jobinfo$行业类别

## (7) 处理观测：公司类别中，非营利机构与事业单位两子类观测过少，没有对比价值，予以删除 ##

table(jobinfo.new$公司类别)

##

## 创业公司 非营利机构 国企 合资 民营公司 上市公司

## 90 20 291 762 4917 406

## 事业单位 外资

## 15 869

jobinfo.new = jobinfo.new[-which(jobinfo.new$公司类别 %in% c("非营利机构", "事业单位")), ]

## (8) 重赋列名 ##

colnames(jobinfo.new) = c("aveSalary", colnames(jobinfo.new[2:13]), "area", "compVar", "compScale", "academic", "exp", "induCate")

## (9) 保存做过预处理的数据集 ##

write.csv(jobinfo.new, file = "数据分析岗位招聘.csv", row.names = FALSE)