```
################################################ 读入数据
################################################
hr<-read.csv("HR.csv")
# 看是否有缺失值
sum(is.na(hr))
summary(hr)
str(hr)
############################################################
############################################################
################################################ 描述性统计
################################################
###### 相关系数矩阵
library(corrplot)
# 给变量重命名
colnames(hr)<-c("满意度","绩效评估","项目数量","每月工作时数","公司工作年数","是否工作事故",
                "是否离职","是否晋升","部门","相对薪资水平")
# 提取数值型变量
HR_correlation1 <- hr[,c(1:6,8,7)]
# 画出相关矩阵图
M1 <- cor(HR_correlation1)
corrplot(M1, method="circle", tl.col = "brown3")

library(ggplot2)
# 将数据赋给 hr_plot，并重命名
hr_plot<-hr
colnames(hr_plot)<-c("满意度","绩效评估","项目数量","平均每月工作时数","在公司工作的年数",
                     "是否发生过工作事故","是否离职","是否在过去五年晋升","部门","相对薪资水平")
# 将是否离职转化为因子型变量
hr_plot$是否离职<-as.factor(hr_plot$是否离职)

attach(hr_plot)
###### boxplot: 连续型变量
# 满意度
p1 <- ggplot(hr_plot, aes(x=是否离职, y=满意度, fill=是否离职)) + geom_boxplot() +
  scale_fill_manual(values=c("brown3", "gray35")) + theme(legend.position="none")
p1

# 绩效评估
p2 <- ggplot(hr_plot, aes(x=是否离职, y=绩效评估, fill=是否离职)) +
```

```
geom_boxplot() +
        scale_fill_manual(values=c("brown3", "gray35")) +
theme(legend.position="none")
p2

# 平均每月工作时数
p3 <- ggplot(hr_plot, aes(x=是否离职, y=平均每月工作时数, fill=是否离职)) +
geom_boxplot() +
        scale_fill_manual(values=c("brown3", "gray35")) +
theme(legend.position="none")
p3

###### barplot: 有序变量
# 项目数量
num_number_project<-as.data.frame(table(项目数量,是否离职))
p4<-ggplot(data=num_number_project, aes(x=项目数量, y=Freq, fill=是否离职,
label=Freq)) +
        labs(x="项目数量", y="个数") + geom_bar(stat="identity") +
        geom_text(data=subset(num_number_project, Freq !=
0),size=3,position =position_stack(vjust=0.5)) +
        guides(fill=guide_legend(title="是否离职")) +
scale_fill_manual(values=c("brown3", "gray35"))
p4

# 在公司工作的年数
num_time_spend_company<-as.data.frame(table(在公司工作的年数,是否离职))
p5<-ggplot(data=num_time_spend_company, aes(x=在公司工作的年数, y=Freq,
fill=是否离职, label=Freq)) +
        labs(x="在公司工作的年数",y="个数") + geom_bar(stat="identity") +
        geom_text(data=subset(num_time_spend_company, Freq != 0), size=3,
position =position_stack(vjust=0.5)) +
        guides(fill=guide_legend(title="是否离职")) +
scale_fill_manual(values=c("brown3", "gray35"))
p5

###### barplot: 类别变量
# 是否有工作事故
num_Work_accident<-as.data.frame(table(是否发生过工作事故,是否离职))
p6<-ggplot(data=num_Work_accident, aes(x=是否发生过工作事故, y=Freq, fill=
是否离职, label=Freq)) +
        labs(x="是否发生过工作事故",y="个数") + geom_bar(stat="identity") +
        geom_text(data=subset(num_Work_accident,Freq != 0), size=3, position
=position_stack(vjust=0.5)) +
        guides(fill=guide_legend(title="是否离职")) +
```

```
scale_fill_manual(values=c("brown3", "gray35"))
p6

# 相对薪资水平
num_salary<-as.data.frame(table(相对薪资水平,是否离职))
p7<-ggplot(data=num_salary,aes(x=reorder(相对薪资水平,-Freq), y=Freq, fill=是否离职, label=Freq)) +
        labs(x="相对薪资水平",y="个数") + geom_bar(stat="identity") + guides(fill=guide_legend(title="是否离职")) +
        geom_text(data=subset(num_salary,Freq != 0),size=3,position =position_stack(vjust=0.5)) +
        scale_fill_manual(values=c("brown3", "gray35"))
p7

# 部门
num_sales<-as.data.frame(table(部门,是否离职))
p8<-ggplot(data=num_sales,aes(x=reorder(部门,-Freq), y=Freq, fill=是否离职, label=Freq)) + labs(x="部门",y="个数") +
        geom_bar(stat="identity") + guides(fill=guide_legend(title="是否离职")) +
        geom_text(data=subset(num_sales,Freq != 0), size=3, position =position_stack(vjust=0.5)) +
        scale_fill_manual(values=c("brown3", "gray35"))
p8

# 部门离职占比
num_sa<-as.data.frame(table(部门,是否离职))
num_sal<-as.data.frame(matrix(1:40,10,4))
colnames(num_sal)<-c("部门","未离职","离职","离职占比")
num_sal$部门<-num_sa[1:10,1]
num_sal$未离职<-num_sa[1:10,3]
num_sal$离职<-num_sa[11:20,3]
num_sal$离职占比<-num_sal$离职/(num_sal$离职+num_sal$未离职)
p8_1<-ggplot(data=num_sal, aes(x=reorder(部门,-离职占比), y=离职占比, fill=部门)) +
        labs(x="部门",y="离职占比") + geom_bar(stat="identity") + scale_fill_manual(values=c(rep(c("gray30","brown3"),2),rep("gray30",6))) +
        theme(legend.position="none", axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))
p8_1

# 是否在过去五年晋升
num_promotion_last_5years<-as.data.frame(table(是否在过去五年晋升,是否离职))
```

```
p9<-ggplot(data=num_promotion_last_5years, aes(x=是否在过去五年晋升, y=Freq, fill=是否离职)) +
        geom_bar(stat="identity", position=position_dodge()) + labs(x="是否在过去五年晋升",y="个数") +
        geom_text(aes(label=Freq), vjust=-0.3, position = position_dodge(0.9), size=3.5) +
        scale_fill_manual(values=c("brown3", "gray35"))
p9

###### 什么样的人会得到晋升？
# 项目数量
num_proj<-as.data.frame(table(项目数量,是否在过去五年晋升))
num_project<-as.data.frame(matrix(1:24,6,4))
colnames(num_project)<-c("项目数量","未晋升","晋升","晋升占比")
num_project$项目数量<-num_proj[1:6,1]
num_project$未晋升<-num_proj[1:6,3]
num_project$晋升<-num_proj[7:12,3]
num_project$晋升占比<-num_project$晋升/(num_project$晋升+num_project$未晋升)
p9_1<-ggplot(data=num_project, aes(x=项目数量, y=晋升占比)) +
        labs(x="项目数量",y="晋升占比") + geom_bar(stat="identity")
p9_1

# 在公司工作的年数
num_year<-as.data.frame(table(在公司工作的年数,是否在过去五年晋升))
num_years<-as.data.frame(matrix(1:32,8,4))
colnames(num_years)<-c("在公司工作的年数","未晋升","晋升","晋升占比")
num_years$在公司工作的年数<-num_year[1:8,1]
num_years$未晋升<-num_year[1:8,3]
num_years$晋升<-num_year[9:16,3]
num_years$晋升占比<-num_years$晋升/(num_years$晋升+num_years$未晋升)
p9_2<-ggplot(data=num_years, aes(x=在公司工作的年数, y=晋升占比, fill=在公司工作的年数)) +
        labs(x="在公司工作的年数",y="晋升占比") + geom_bar(stat="identity") +
        scale_fill_manual(values=rep("brown3",8)) + theme(legend.position="none")
p9_2

# 部门
num_pro<-as.data.frame(table(部门,是否在过去五年晋升))
num_promotion<-as.data.frame(matrix(1:40,10,4))
colnames(num_promotion)<-c("部门","未晋升","晋升","晋升占比")
num_promotion$部门<-num_pro[1:10,1]
```

```
num_promotion$未晋升<-num_pro[1:10,3]
num_promotion$晋升<-num_pro[11:20,3]
num_promotion$ 晋 升 占 比 <-num_promotion$ 晋 升 /(num_promotion$ 晋 升 +num_promotion$未晋升)
p9_3<-ggplot(data=num_promotion, aes(x=reorder(部门,-晋升占比), y=晋升占比, fill=部门)) +
        labs(x=" 部 门 ",y=" 晋 升 占 比 ") + geom_bar(stat="identity") + scale_fill_manual(values=rep("brown3",10)) +
        theme(legend.position="none",
axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))
p9_3

# 相对薪资水平
num_sal<-as.data.frame(table(相对薪资水平,是否在过去五年晋升))
num_salary<-as.data.frame(matrix(1:12,3,4))
colnames(num_salary)<-c("相对薪资水平","未晋升","晋升","晋升占比")
num_salary$相对薪资水平<-num_sal[1:3,1]
num_salary$未晋升<-num_sal[1:3,3]
num_salary$晋升<-num_sal[4:6,3]
num_salary$晋升占比<-num_salary$晋升/(num_salary$晋升+num_salary$未晋升)
p9_4<-ggplot(data=num_salary, aes(x=reorder(相对薪资水平,-晋升占比), y=晋升占比)) +
        labs(x="相对薪资水平",y="晋升占比") + geom_bar(stat="identity")
p9_4
detach(hr_plot)
############################################################
############################################################

################################################## 数据预处理 ##############################################
# 选出有价值的员工
hr<-subset(hr,绩效评估>=0.7 | 项目数量>=6 | 公司工作年数>=4)
# 将类别变量的类型转化为因子型
hr$是否离职<-as.factor(hr$是否离职)
hr$是否工作事故<-as.factor(hr$是否工作事故)
hr$是否晋升<-as.factor(hr$是否晋升)
hr1<-hr
hr2<-hr
# 将用于 logistic 回归的数据中的连续型变量标准化处理
hr1$满意度<-scale(hr1$满意度)
hr1$绩效评估<-scale(hr1$绩效评估)
hr1$项目数量<-scale(hr1$项目数量)
hr1$每月工作时数<-scale(hr1$每月工作时数)
```

```
hr1$公司工作年数<-scale(hr1$公司工作年数)
#########################################################
#########################################################

#################################################  建立模型 ################################################
###### Logistic 回归
#  模型拟合
glm.fit<-glm(是否离职~., data = hr1, family = "binomial")
summary(glm.fit)
#  系数可视化
glm.coef <- as.data.frame(glm.fit$coefficients) #  提取回归系数
colnames(glm.coef)<-"系数" #  变量重命名
glm.coef$变量  = rownames((glm.coef)) #  添加一个名为“变量”的变量
glm.coef$显著性<-rep("",19) #  添加一个名为“显著性”的变量
pvalue<-summary(glm.fit)$coef[,4] #  根据 p 值给出显著性
for(i in 1:19){
        if(pvalue[i]<10e-4){glm.coef$显著性[i]="***"}
        else
                if(10e-4<pvalue[i]  &  pvalue[i]<0.01){glm.coef$ 显 著 性 [i]="**"}
        else
                if(0.01<pvalue[i] & pvalue[i]<0.1){glm.coef$显著性[i]="."}
}
glm.coef1 <- glm.coef[9:17,] #  提取部门系数
glm.coef2 <- glm.coef[c(2:8,18,19),] #  提取其余变量的系数
p10  <-  ggplot(data=glm.coef1,  aes(x=reorder( 变 量 ,- 系 数 ),  y= 系 数 ))  + geom_bar(aes(fill=变量), stat="identity") +
        labs(x="",y="  系  数  ")  +  theme(legend.position="none", axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)) +
        geom_text(size=8,data=glm.coef1,aes(x=变量, y=系数,label=显著性)) +

scale_fill_manual(values=c("brown3","gray35","brown3",rep("gray35",6)))
p10
p11  <-  ggplot(data=glm.coef2,  aes(x=reorder( 变 量 ,- 系 数 ),  y= 系 数 ))  + geom_bar(aes(fill=变量), stat="identity") +
        labs(x="",y="  系  数  ")  +  theme(legend.position="none", axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)) +
        geom_text(size=8,data=glm.coef2,aes(x=变量, y=系数,label=显著性)) +

scale_fill_manual(values=c(rep("gray35",5),rep("brown3",2),rep("gray35",2)))
p11

######  决策树
```

```
library(rpart)
library(rattle)
# 模型拟合
rpart.fit<- rpart(是否离职~., data = hr2)
summary(rpart.fit)
# 画出分类树
fancyRpartPlot(rpart.fit, cex=0.8, palettes=c("Greys", "Reds"))

###### 随机森林
library(randomForest)
# 模型拟合
set.seed(1234)
random.fit<-randomForest(是否离职~., data=hr2, ntree=100, importance=TRUE)
#100 棵树
summary(random.fit)
# 画出变量重要性图
random.importance<-as.data.frame(importance(random.fit)[,4]) # 提取各个变量的 Gini 减少量
colnames(random.importance)<-"Gini 减少量" # 变量重命名
random.importance$变量<-rownames(random.importance) # 添加一个名为“变量”的变量
p12 <- ggplot(data=random.importance, aes(x=reorder(变量,Gini 减少量), y=Gini 减少量, fill=变量, label=Gini 减少量)) +
        labs(x="") + geom_bar(stat="identity") +
theme(legend.position="none") + coord_flip() +

scale_fill_manual(values=c(rep("gray35",3),"brown3",rep("gray35",2),"brown3",
rep("gray35",2)))
p12
##########################################################
##########################################################

################################################ 模型比较 ################################################
###### 全样本曲线 ROC
library(pROC)
# 用估计的模型对全样本重新进行预测
glm.probs=predict(glm.fit,hr1,type="response")
rpart.probs<-predict(rpart.fit,hr2,type="prob")[,2]
random.probs<-predict(random.fit,newdata=hr2,type="prob")[,2]

glm_roc <- roc(hr1$是否离职,glm.probs)
rpart_roc <- roc(hr2$是否离职,rpart.probs)
random_roc <- roc(hr2$是否离职,random.probs)
```

```
plot(glm_roc, lty=1, main="全样本的三个模型的 ROC 曲线对比", lwd=3,
col="brown2") # AUC=0.9284
plot(rpart_roc,add=T, col="steelblue4", lty=2, lwd=3) # AUC=0.9401
plot(random_roc,add=T, col="darkorange2", lty=2, lwd=3) # AUC=1
legend(0.5,0.4,c("Logistic","Decision Tree","Random Forest") , cex=0.8,
       col=c("brown2","steelblue4","darkorange2"), lty=c(1,2,2), lwd=3)

###### 比较各个模型精度
### 定义计算精度的函数
# Logistic
confusion <- function(tag,prob,p,n){
        re <- rep("real_NO",n)
        re[tag==1] <- "real_Yes"
        pred <- rep("NO",n)
        pred[prob>p]="Yes"
        return(table(as.vector(pred),as.vector(re)))
} # 定义混淆矩阵
accuracy_logistic <- function(Y,prob,p,n){
        sensitivity <- NULL
        specificity <- NULL
        total_accuracy <- NULL
        temp <- confusion(Y,prob,p,n)
        sensitivity <- temp[2,2]/(temp[1,2]+temp[2,2])
        specificity <- temp[1,1]/(temp[1,1]+temp[2,1])
        total_accuracy <- (temp[1,1]+temp[2,2])/n
        accuracy                                                    <-
c(sensitivity=sensitivity,specificity=specificity,total_accuracy=total_accuracy)
        return(accuracy)
}
# 求 Logistic 的最优阈值
plot(glm_roc,print.auc=T,print.auc.x=0.4,print.auc.y=0.4,print.thres=T,
     print.auc.cex=1.5,print.thres.cex=1.5,main="ROC of lotistic_train") #
p=0.313,AUC=0.9284

# 决策树 & 随机森林
accuracy_other <- function(fit,data,n){
        sensitivity <- NULL
        specificity <- NULL
        total_accuracy <- NULL
        probs<-predict(fit,data,type="class")
        temp<- table(probs,data$是否离职)
        sensitivity <- temp[2,2]/(temp[1,2]+temp[2,2])
        specificity <- temp[1,1]/(temp[1,1]+temp[2,1])
```

```
        total_accuracy <- (temp[1,1]+temp[2,2])/n
        accuracy <- c(sensitivity=sensitivity,specificity=specificity,total_accuracy=total_accuracy)
        return(accuracy)
}

### 五折交叉验证
set.seed(123)
# 将样本随机分成大小相等的五份
n<-sample(nrow(hr))
hr1<-hr1[n,]
hr2<-hr2[n,]
folds <- cut(seq(1,nrow(hr)),breaks=5,labels=FALSE)
# 定义六个空的数据框用于后面存储结果
a11<-data.frame()
a12<-data.frame()
a21<-data.frame()
a22<-data.frame()
a31<-data.frame()
a32<-data.frame()
for(i in 1:5){
        testIndexes <- which(folds==i,arr.ind=TRUE)
        # Lotistic
        testData1 <- hr1[testIndexes, ]
        trainData1 <- hr1[-testIndexes, ]
        glm.fit <- glm(是否离职~., data = trainData1, family = "binomial")
        glm.train.probs <- predict(glm.fit,trainData1,type="response") # 训练集预测结果
        glm.test.probs <- predict(glm.fit,testData1,type="response") # 测试集预测结果
        a11[i,1]<-accuracy_logistic(trainData1$是否离职,glm.train.probs,0.313,nrow(trainData1))[1] # sensitivity
        a11[i,2]<-accuracy_logistic(trainData1$是否离职,glm.train.probs,0.313,nrow(trainData1))[2] # specificity
        a11[i,3]<-accuracy_logistic(trainData1$是否离职,glm.train.probs,0.313,nrow(trainData1))[3] # total_accuracy
        a12[i,1]<-accuracy_logistic(testData1$是否离职,glm.test.probs,0.313,nrow(testData1))[1] # sensitivity
        a12[i,2]<-accuracy_logistic(testData1$是否离职,glm.test.probs,0.313,nrow(testData1))[2] # specificity
        a12[i,3]<-accuracy_logistic(testData1$是否离职,glm.test.probs,0.313,nrow(testData1))[3] # total_accuracy
        # 决策树
        testData2 <- hr2[testIndexes, ]
```

```
        trainData2 <- hr2[-testIndexes, ]
        rpart.fit<- rpart(是否离职~., data=trainData2)
        a21[i,1]<-accuracy_other(rpart.fit,trainData2,nrow(trainData2))[1]
        a21[i,2]<-accuracy_other(rpart.fit,trainData2,nrow(trainData2))[2]
        a21[i,3]<-accuracy_other(rpart.fit,trainData2,nrow(trainData2))[3]
        a22[i,1]<-accuracy_other(rpart.fit,testData2,nrow(testData2))[1]
        a22[i,2]<-accuracy_other(rpart.fit,testData2,nrow(testData2))[2]
        a22[i,3]<-accuracy_other(rpart.fit,testData2,nrow(testData2))[3]
        # 随机森林
        set.seed(1234)
        random.fit<-randomForest(是否离职~., data=trainData2, ntree=100,
importance=TRUE) #100 棵树
        a31[i,1]<-accuracy_other(random.fit,trainData2,nrow(trainData2))[1]
        a31[i,2]<-accuracy_other(random.fit,trainData2,nrow(trainData2))[2]
        a31[i,3]<-accuracy_other(random.fit,trainData2,nrow(trainData2))[3]
        a32[i,1]<-accuracy_other(random.fit,testData2,nrow(testData2))[1]
        a32[i,2]<-accuracy_other(random.fit,testData2,nrow(testData2))[2]
        a32[i,3]<-accuracy_other(random.fit,testData2,nrow(testData2))[3]
}

# 计算各个精度的均值
sensitivity_train<-c(mean(a11[,1]),mean(a21[,1]),mean(a31[,1]))
sensitivity_train
sensitivity_test<-c(mean(a12[,1]),mean(a22[,1]),mean(a32[,1]))
sensitivity_test
specificity_train<-c(mean(a11[,2]),mean(a21[,2]),mean(a31[,2]))
specificity_train
specificity_test<-c(mean(a12[,2]),mean(a22[,2]),mean(a32[,2]))
specificity_test
total_train<-c(mean(a11[,3]),mean(a21[,3]),mean(a31[,3]))
total_train
total_test<-c(mean(a12[,3]),mean(a22[,3]),mean(a32[,3]))
total_test
##########################################################
##################################################

################################################## 雷达图
##################################################
library(fmsb)
# 定义计算众数的函数
getmode <- function(v) {
        uniqv <- unique(v)
        uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
# 分别提取每个维度的最大值和最小值组成两个样本
maxmin <- data.frame(
        满意度=c(max(hr$满意度),min(hr$满意度)),
        公司工作年数=c(max(hr$公司工作年数),min(hr$公司工作年数)),
        每月工作时数=c(max(hr$每月工作时数),min(hr$每月工作时数)),
        绩效评估=c(max(hr$绩效评估),min(hr$绩效评估)),
        项目数量=c(max(hr$项目数量),min(hr$项目数量)))
# 提取离职员工的样本
hr_left<-subset(hr,是否离职==1)
# 构建一个离职员工的样本
dat <- data.frame(
        满意度=mean(hr_left$满意度), # 均值
        公司工作年数=getmode(hr_left$公司工作年数), # 众数
        每月工作时数=mean(hr_left$每月工作时数), # 均值
        绩效评估=mean(hr_left$绩效评估), # 均值
        项目数量=getmode(hr_left$项目数量)) # 众数
#合并三个样本
dat <- rbind(maxmin,dat)
# 画雷达图
radarchart(dat, axistype=1, pcol="#CD3333", plwd=3, cglcol="grey", cglty=1,
           axislabcol="grey", cglwd=0.8, vlcex=0.8 )
##########################################################
##########################################################
```