一、题目

结合本课程所学知识，从你熟悉的应用问题中获取数据，并使用以下知识点对数据进行分析、预测或者聚类（不得使用课堂所讲案例），形成分析报告。

选题中必须使用的知识点

1. 对数据的规范化处理（正态归一化、分位点）。
2. 分类与预测方法，包括Fisher线性判别法、决策树方法、支撑向量机SVM方法。
3. 聚类方法，包括kmeans，层次聚类等。
4. 离群点分析与异常检测方法, 包括基于箱底规则的方法、基于线性回归的方法、基于离群因子的方法、基于聚类的方法等。
5. 社会网络分析方法, 包括网络群集发现、三角形侦测。鼓励选用教师提供的海关数据。
6. 时序数据分析与预测，包括股票分析技术。
7. 主成分分析方法
8. 关联规则挖掘算法，包括Apriori算法

二、数据分析报告的内容

1. 选题背景
2. 数据来源与数据描述
3. R语言代码
4. 运行结果（要求包括有图表和结论）。

# 三、成绩评定

第18周以小组分别答辩的形式，每组同学派一位代表对选题背景和数据来源进行说明，并进行程序演示，由教师进行评定，小组评定成为小组内同学的期末成绩

## 一、选题背景

联合国有关报告指出，人口老龄化是人类历史上前所未有的一场无声的革命，足以影响或改变未来。毫无疑问，人口老龄化将成为我国基本国情的组成部分，成为我国实现两个一百年奋斗目标进程中必须考虑的一个长期的变量。

我国人口老龄化的影响

1、人口老龄化对经济增长的影响

首先，老龄化直接导致劳动力供给的减少，潜在经济增长率面临下行的压力。明显影响制造业出口利润。其次，老龄化使人口结构老化，消费能力下降，导致需求不足。在此情况下，政府财政面临收入减缓和养老、医疗等社会保障支出增加的双重压力，将严重制约政府直接投资和间接引导投资的能力。

2、人口老龄化对结构调整的影响

研究表明，随着年龄的增长，劳动力个体的生产率呈现先增后降的变化态势，一般是在40岁达到顶峰，此后明显下降。今后一个时期，我国劳动年龄人口中位数将快速上升，到2030年45岁以下的劳动年龄人口将减少四分之一，劳动力将整体老化，将对结构调整产生不利影响。

3、人口老龄化对创新驱动的影响

新常态下，由要素驱动、投资驱动转向创新驱动，是必由之路。因此，只能从全要素生产率上做文章，特别是在创新上做文章。而人口劳动力的老化将影响社会活力，明显弱化全社会创新创造能力。从需求方面看，消费对经济增长的重要性会越来越大，而人口老龄化可能不利于消费领域新产品、新技术的应用和推广，年龄越大，对新技术的接受能力就差一些，不愿意接受新生事物。

4、人口老龄化对社会保障的影响

首先，老年人口增加将使政府公共财政在基本养老金、退休金、保险费补贴和医疗方面的支出增加。其次，人口老龄化改变家庭结构和规模，抚养比上升，削弱家庭的养老功能，导致家庭养老的风险增加，家庭的代际矛盾显性化

了解完相关资料之后我们的感想

所以说老龄化的研究和预测不仅是对国家而且对个人都有着十分重要的意义

我们了解了影响中国老龄化的主要因素。计量结果显示，男女比例，人口总数，经济发展、人口自然增长率下降、城镇化程度提高和医疗服务水平改善等

我们希望通过选取与老龄化相关的指标预测2021人口老龄化比率

## 二、实验思路

1、选取与老龄化相关的指标

2、查找数据，并根据相似的观测值填补缺失值

3、使用pairs()方法绘制各指标与老龄化比率的散点图，观测是否存在相关性，剔除无关指标

4、进行基于线性回归的离群点检测并去除离群点

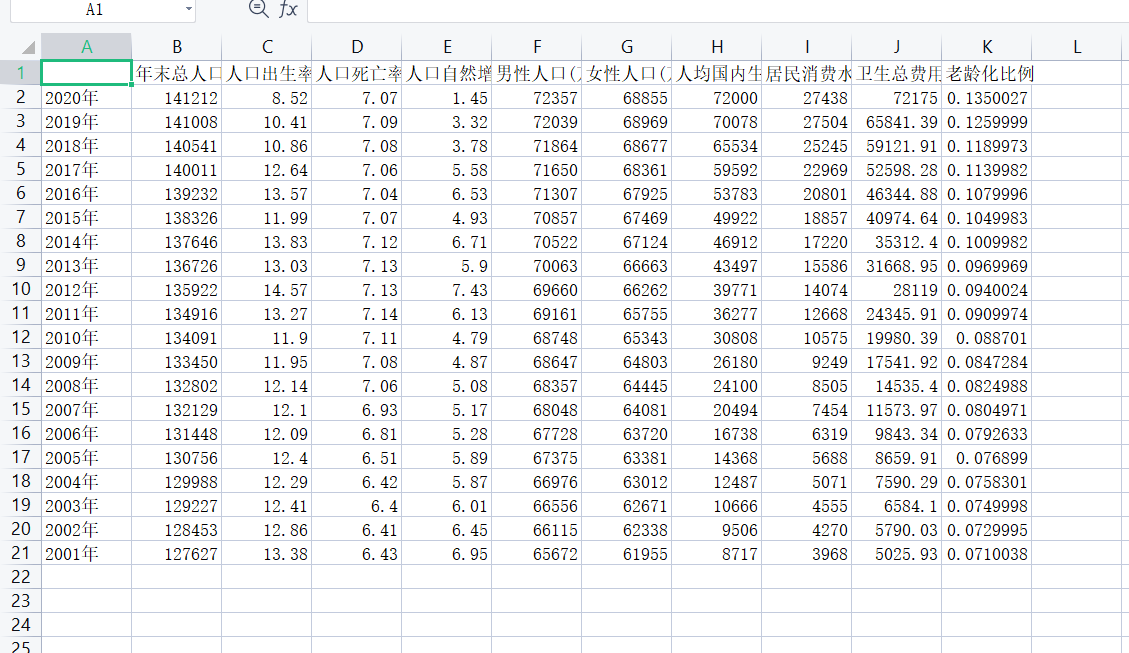
5、进行基于fisher线性判别法的对2021年末人口老龄化比例的预测，得出结果

6、进行基于随机森林的2021年末人口老龄化比例的预测，得出结果

## 三、数据来源

我们在国家统计局中查找数据，并根据相似的观测值填补缺失值，得到的实验数据如下图所示，可以看出我们选择了年末人口总数，人口出生率，人口自然增长率，人口死亡率，男性人口，女性人口，人均国内生产总值，居民消费水平，卫生总费用这九个指标，希望通过这九个指标预测人口老龄化比率。

数据如图1所示：



**图1：实验的源数据**

## 四、实验的主要内容

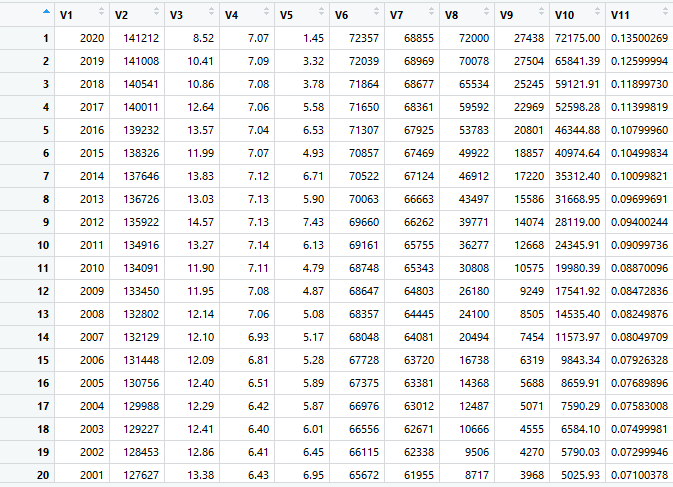
### 1、数据导入

将数据导入Rstudio中，代码如下所示：

**data<-read.csv("C:\\Users\\lenovo\\Desktop\\R语言\\数据.csv",header=FALSE,sep=",")#读取数据**

**data**

导出的数据如图2所示：



**图2：将数据导入RStudio中**

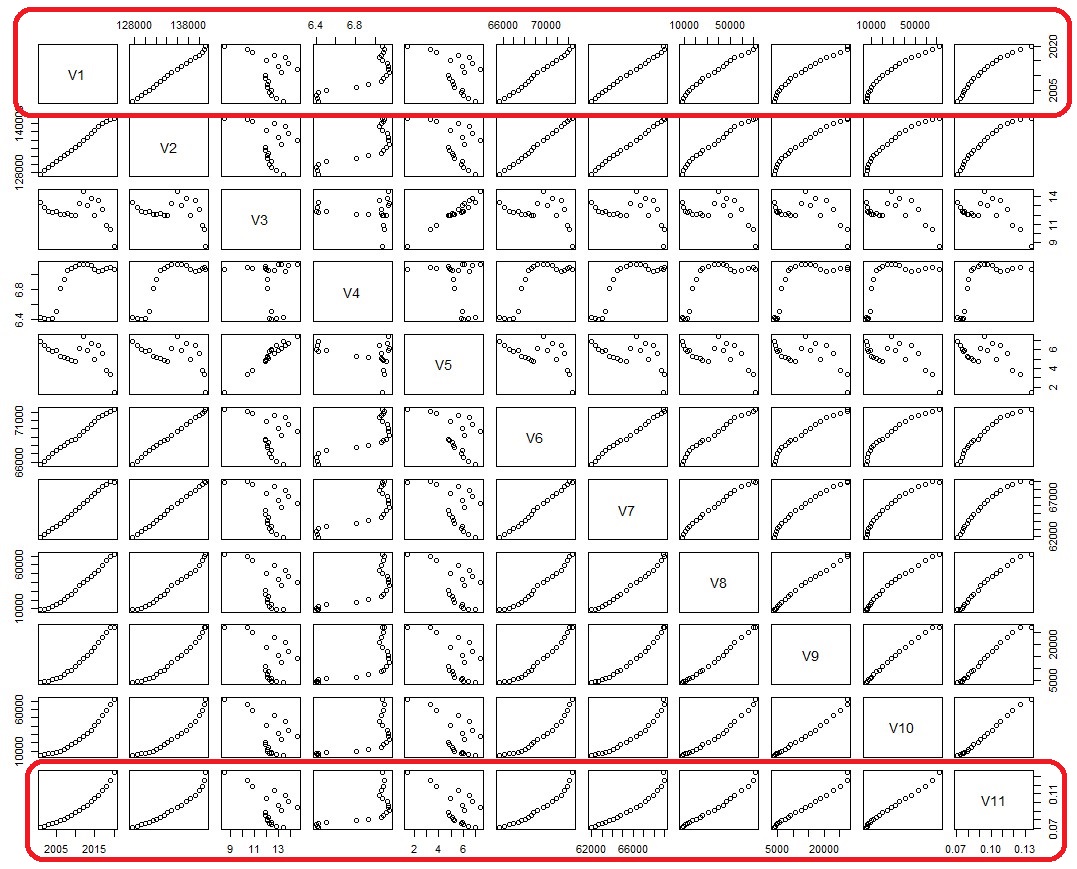
导入的数据为csv格式，用逗号隔开，由于类名过于复杂，在对数据的处理中我们不直接使用列名，而是使用系统自排序的V序列列名。

### 2、相关性检测

用pairs方法绘制散点图矩阵来查看各指标与人口老龄化的关系，并去除无关指标，代码如下所示：

**pairs(data[,2:11])**

pairs方法显示的各指标相关性如图3所示：



**图3：pairs方法查看各指标与人口老龄化比例的相关性**

从图3中可以看出v3,v4,v5相关性最差，所以去指标v3,v4,v5 ，代码如下所示：

**data=data[,-3:-5]**

图中最后一行，即各指标与V11的相关性，便是与人口老龄化比例的相关性，从图中可以看出，各指标与人口老龄化比例有着很强的线性关系，且大都是正相关的关系，因此本次实验可以建立线性模型来进行预测，并且，从图中第一行可以看到，大部分指标与V1，即与年份变化存在很强的线性关系，因此在进行2021年各指标数据预测中使用线性回归预测会相对而言较为合适。

### 3、建立线性回归模型fit，并对原数据各年份进行老龄化预测

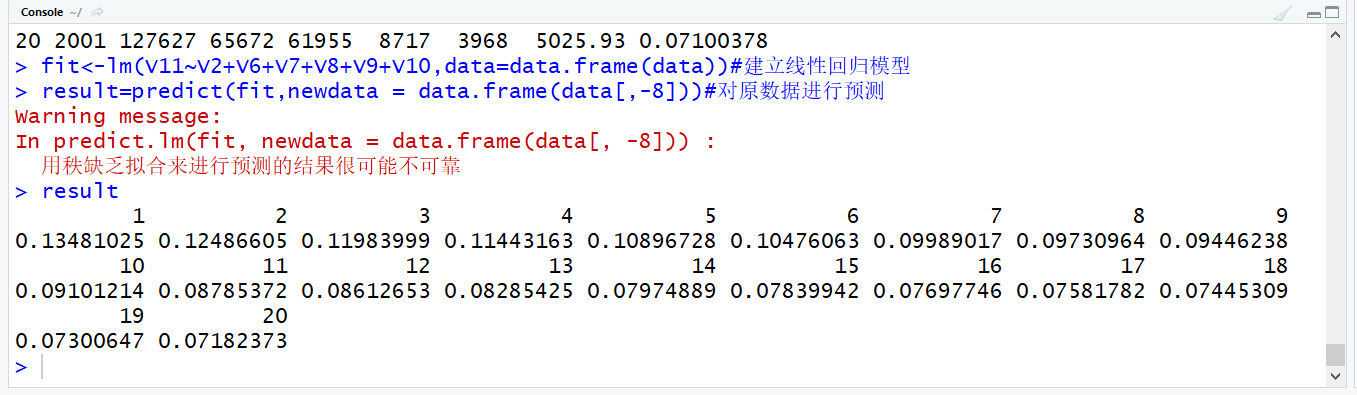
从相关性检测中可以发现，各指标与年时间序列，以及人口老龄化比例之间，存在很强的线性相关性，因此，为预测2021年各指标数据，使用线性回归模型将较为合适。并且，为检测线性回归模型的可用性，我们决定使用该模型对原数据进行测试。代码如下所示：

**fit<-lm(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data))**

**result=predict(fit,newdata = data.frame(data[,-8]))**

**result**

该线性模型通过V2，V6，V7，V8，V9，V10六指标对人口老龄化比例进行预测，预测的结果如图4所示



**图4：原数据的预测结果**

### 4、计算预测的误差率

由于上图中预测时提示该预测结果可能不可靠，因此有必要对预测结果进行误差率计算，通过平均误差率的大小判断该模型是否足够准确，在计算过程中，我们将真实值与预测值的差占真实值的比例定义为误差率，再求出20条数据的平均误差率avg，代码如下所示：

**sum=0**

**for (i in 1:20) {**

**err\_rate=(data[i,8]-result[i])/data[i,8]**

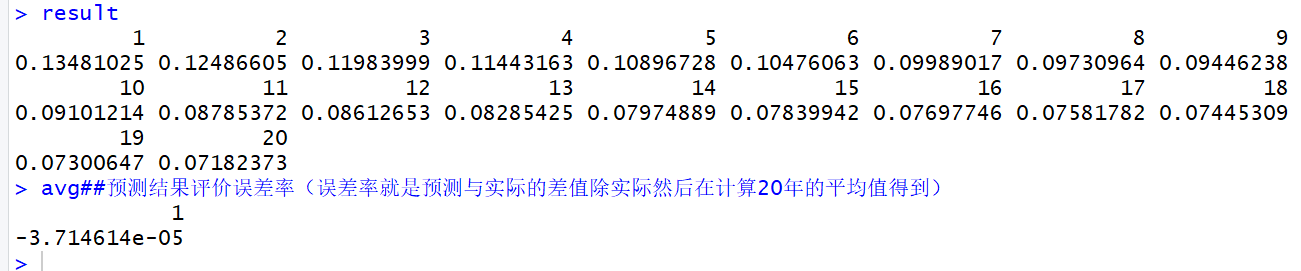
**sum=sum+err\_rate**

**}**

**avg=sum/20**

**avg**

得出的平均误差率avg结果如图5所示



**图5：平均误差率的计算**

得出平均误差率为3.71\*10e-5，极小，所以该预测方法较为准确；

### 5、基于线性回归的离群点检测

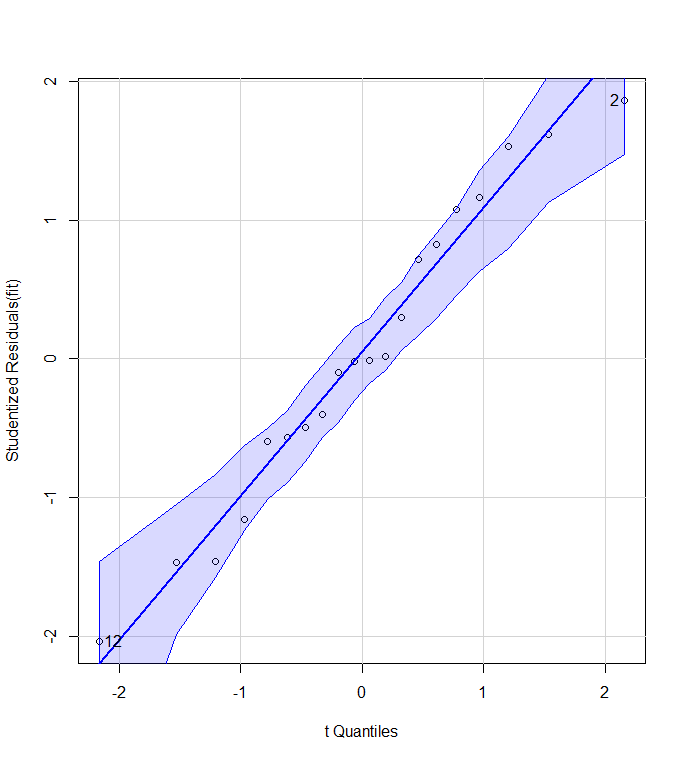
由于我们采用的数据为2001年至2020年各指标的数据，如果在此期间发生了某些不可抗力的因素，可能导致该年份的各指标数据脱离原变化趋势，但是我们需要计算的是正常情况下的各指标2021年数据，因此需要将出现误差的数据点剔除，因此有必要进行离群点检测，且由于之前以及建立了线性回归模型，可以利用该模型进行基于线性回归的离群点检测。

代码如下所示：

**library("car")**

**qqPlot(fit,labels=rownames(data), envelope=TRUE)**

导入“car”包，调用“car”包的qqPlot()函数，对fit线性回归模型进行离群点检测，将置信区间设置为0.89，运行代码，结果如图6所示，图中每个点是每一列数据的点；



**图6：离群点分布**

从图中可以看出，qqPlot()方法找出了最离群的两个数据点，分别为2和12，即2009年与2019年的数据，但是，可以发现这两个点依旧属于可利用范围内，也就是说，这两个点并未发生太大的偏差，因此我们依旧将其归入可利用数据范畴，本次离群点检测无明显离群点，不剔除数据。

### 6、数据的预定义

由于我们在对2021年人口老龄化比例线性预测过程中需要将2021年各指标数据插入原表，因此在预测前我们需要对部分数据进行预定义，按照原表格式定义data2021，即第一个属性为年份即2021，中间六个属性为六个判断指标的值，暂时将其设置为0，最后一个为人口老龄化比例的值，也暂时设置为0.并且因为我们对各判断指标2021年数据的预测是基于其和年时间序列的强线性相关性进行的线性回归预测，因此事先定义一个2001至2020的年时间序列year。代码如下所示：

**data2021=c(2021,0,0,0,0,0,0,0)##数据的预定义**

**year=rep(2020:2001)**

### 7、基于线性回归的各指标2021年数据预测

从相关性检测中可以发现，六个判断指标与年时间序列有着很强的线性相关性，因此可以使用线性回归模型进行预测，代码如下所示：

**for (i in 2:7) {**

**lm=lm(data[,i]~year)**

**data2021[i]=predict(lm,newdata = data.frame(year=2021))**

**}**

我们通过一个循环结构对数据集2~7列数据也就是六个判断指标进行基于线性回归的预测，并将其值赋予预定义的data2021数据集，预测结果如图7所示



**图7：各指标2021年数据预测结果**

### 8、对预测的2021年数据进行基于fisher线性判别法的预测

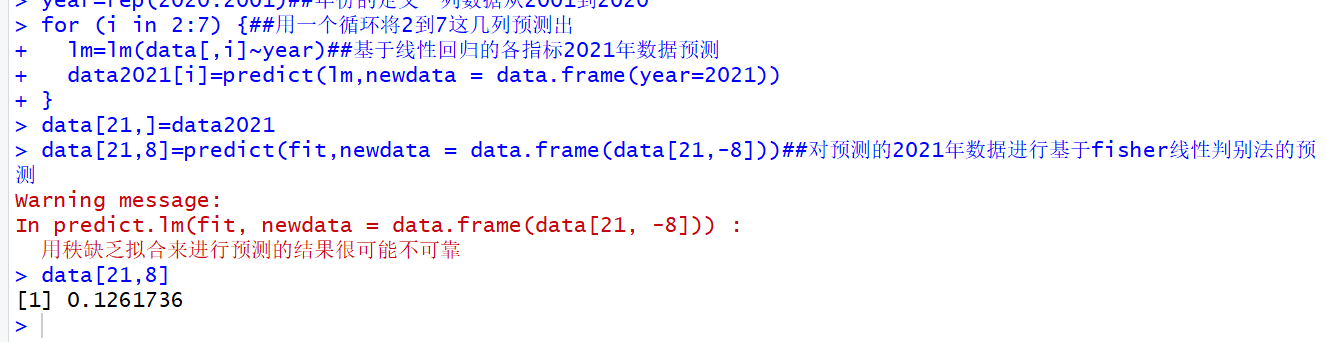
此时，我们以及拥有了各判断指标2021年的数据，现在，可以利用初始建立的线性回归模型fit进行2021年人口老龄化比例的预测，代码如下所示：

**data[21,]=data2021**

**data[21,8]=predict(fit,newdata = data.frame(data[21,-8]))##**

**data[21,8]**

我们将预定义的data2021数据集插入原始数据集中，并将2021年人口老龄化比例的预测结构放入data[21,8]中，结果如图8所示：



**图8：2021年人口老龄化比例预测结果**

经过线性回归模型fit的预测，2021年人口老龄化比例约为12.61%。

### 建立随机森林模型

到此我们已经对2021年末人口老龄化比例进行了基于Fisher线性判别的预测，且得出了预测结果，但是为确保预测结果的有效性，现在对2021年末人口老龄化比例进行基于随机森林的预测，通过比较两次预测的结果可以得出一个更有说服力的结论。

下面我们建立随机森林模型，代码如下所示：

**install.packages("randomForest")**

**library("randomForest")**

**set.seed(1234)**

**rfmode<randomForest(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data),importance=TRUE,proximity=TRUE)**

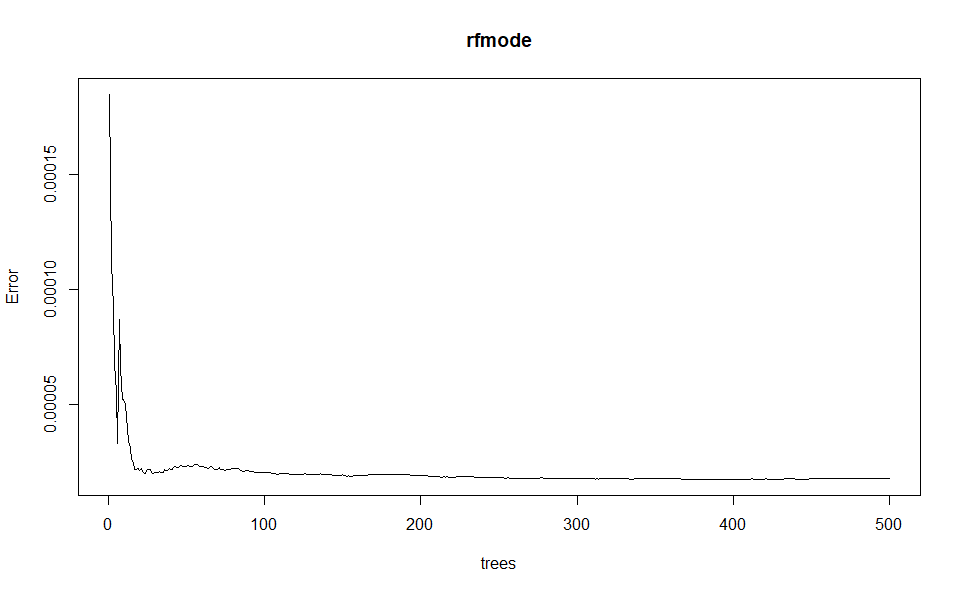
我们安装并引入随机森林所需的“randomForest”包，并设置随机种子数，确保以后再执行代码时可以得到一样的结果，通过randomForest()方法建立模型，该模型同样是通过V2，V6，V7，V8，V9，V10六指标对人口老龄化比例进行预测。

### 改进随机森林模型

由于随机森林并不是决策树越多越好，为了找到使模型最优的决策树数量，使用plot()方法得出该模型误差率与决策树数量的关系图，代码如下所示：

**plot(rfmode)**

绘制结果如图9所示：



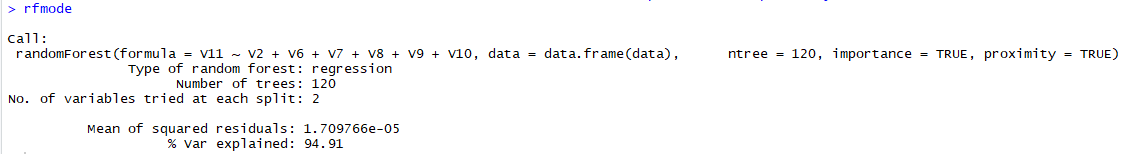
**图9：随机森林模型决策树数量与误差率的关系图**

从图中可以看出，在树的数量约大于120的时候，基本处于稳定状态，在保证效能的情况下减少决策树的数量，减少运行时间，在这里我们就去决策树数量为120，重新建立随机森林模型，代码如下所示：

**rfmode<randomForest(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data),ntree=120,importance=TRUE,proximity=TRUE)**

增加了一个参数ntree，并设置为120.

模型的基本信息如图10所示：



**图10：rfmode模型的基本信息**

从中可以看出，基于方差的总体解释率var explained为94.91%，相当可观的一个数值，表明人口老龄化比例与各判断指标密切相关。

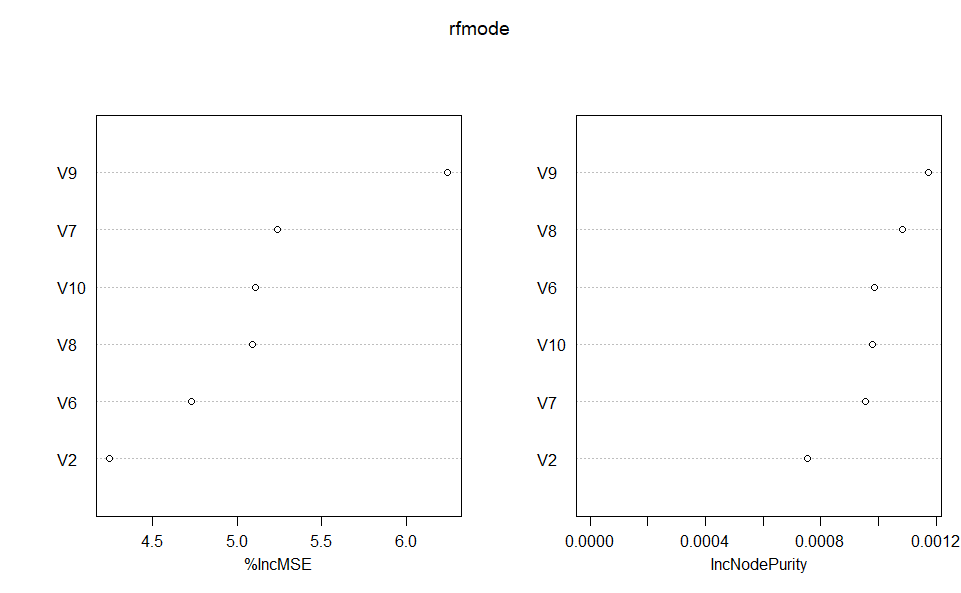
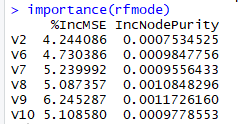
### 评估判断指标的重要性

基于已经构建好的随机森林回归模型，可以从中评估各判断指标的重要性，代码如下所示：

**importance(rfmode)**

**varImpPlot(rfmode)**

运行代码，结果如图11所示：



**图11：各判断指标的重要性排序**

“%IncMSE”即increase in mean squared error，通过对每一个预测变量随机赋值，如果该预测变量更为重要，那么其值被随机替换后模型预测的误差会增大。因此，该值越大表示该变量的重要性越大；

“IncNodePurity”即increase in node purity，通过残差平方和来度量，代表了每个变量对分类树每个节点上观测值的异质性的影响，从而比较变量的重要性。该值越大表示该变量的重要性越大。

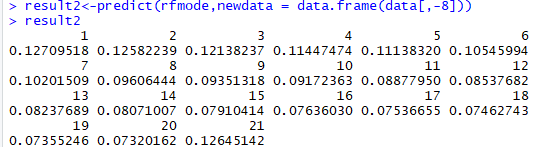
### 12、进行基于随机森林模型的各年份人口老龄化比例预测

我们的随机森林模型以及建立完成，接下来同样将其应用到我们的数据中，对各年份人口老龄化比例进行预测，代码如下所示：

**result2<-predict(rfmode,newdata = data.frame(data[,-8]))**

**result2**

预测结果如图12所示：



**图12：基于随机森林的人口老龄化比例预测结果**

从图中可以看出，第21位数据，即2021年人口老龄化比例为12.64%。

同样的，我们对2001年至2020年的预测结果计算平均误差率，代码如下所示：

**sum=0**

**for (i in 1:20) {**

**err\_rate=(data[i,8]-result2[i])/data[i,8]**

**sum=sum+err\_rate**

**}**

**avg=sum/20**

**avg**

该计算类似于线性回归模型时的平均误差率计算，因此不多赘述，预测结果，平均误差率avg如图13所示：



**图13：随机森林模型预测的平均误差率**

从图中可知，随机森林模型的平均误差率avg为-0.001567248，虽不如线性回归模型的平均误差率小，但依旧比较可观。

### 双模型预测结果比较

在Fisher线性判别模型fit的预测中，得出2021年末人口老龄化比例约为12.61%，而在随机森林模型rfmode的预测中，得出2021年末人口老龄化比例约为12.64%，两者之间的差距为0.03%，可以说极为接近，因此此次实验预测结果较为准确。

## 四、总结与改进

总结：预测得出，2021年，中国人口老龄化比例约为12.6%，较2020年的13.5%，略微降低，分析原因，发现在对V10即卫生总费用指标的预测较2020年降低幅度较大，因此导致预测结果较低。

从中可以发现，若加大对卫生总费用的投入，可以有效的降低人口老龄化的比例。

改进：

1、数据较少：若建立模型的数据足够多，模型将更加准确

2、涉及指标较少：影响人口老龄化的因素涉及方方面面，此模型只考虑了其中九个，若增加判断指标，可以优化模型。

**R语言完整代码：**

data<-read.csv("C:\\Users\\lenovo\\Desktop\\R语言\\数据.csv",header=FALSE,sep=",")

data

pairs(data[,1:11])

data=data[,-3:-5]

data

fit<-lm(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data))

result=predict(fit,newdata = data.frame(data[,-8]))

result

sum=0

for (i in 1:20) {

err\_rate=(data[i,8]-result[i])/data[i,8]

sum=sum+err\_rate

}

avg=sum/20

avg

library("car")

qqPlot(fit,labels=rownames(data) ,envelope=TRUE)

data2021=c(2021,0,0,0,0,0,0,0)

year=rep(2020:2001)

for (i in 2:7) {

lm=lm(data[,i]~year)

data2021[i]=predict(lm,newdata = data.frame(year=2021))

}

data[21,]=data2021

data[21,8]=predict(fit,newdata = data.frame(data[21,-8]))

data[21,8]

install.packages("randomForest")

library("randomForest")

set.seed(1234)

rfmode<-randomForest(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data),importance=TRUE,proximity=TRUE)

plot(rfmode)

rfmode<-randomForest(V11~V2+V6+V7+V8+V9+V10,data=data.frame(data),ntree=120,importance=TRUE,proximity=TRUE)

rfmode

importance(rfmode)

varImpPlot(rfmode)

result2<-predict(rfmode,newdata = data.frame(data[,-8]))

result2

sum=0

for (i in 1:20) {

err\_rate=(data[i,8]-result2[i])/data[i,8]

sum=sum+err\_rate

}

avg=sum/20

avg