关联规则挖掘

1. 数据源

数据集：San Francisco Building Permits

下载地址： [地址](https://pan.baidu.com/s/1rayD18hDMD8elzcE9GYs1g)

数据集中属性解释：见下载地址中DataDictionaryBuildingPermit.xlsx

2. 要求

1. 对数据集进行处理，转换成适合关联规则挖掘的形式；
2. 找出频繁项集；
3. 导出关联规则，计算其支持度和置信度
4. 对规则进行评价，可使用Lift，也可以使用教材中所提及的其它指标

3. 实验过程

#### 3.1 数据预处理

San Francisco Building Permits 数据集中总共包含43个属性，实验之前先对进行关联规则挖掘的属性做了筛选。类似于类型描述（如Proposed.Construction.Type.Description）属性、日期属性（如Filed.Date）、仅有一种属性值的属性（如Structural.Notification）都被舍弃。最终确定13中属性参与关联规则挖掘，它们分别是：Permit.Type, Street.Number, Unit, Current.Status, Number.of.Existing.Stories, Number.of.Proposed.Stories, Estimated.Cost, Revised.Cost, Existing.Units, Proposed.Units, Existing.Construction.Type,

Proposed.Construction.Type, Supervisor.District。用以上13中属性构建新的待处理数据集，代码如下：

*data = read.csv('Building\_Permits.csv', header = T) #读取原数据集*

*library("dplyr") #加载“dplyr”包，用于做关联规则挖掘*

*data = select(data, Permit.Type, Street.Number, Unit, Current.Status,*

*Number.of.Existing.Stories, Number.of.Proposed.Stories, Estimated.Cost, Revised.Cost,*

*Existing.Units, Proposed.Units, Existing.Construction.Type,*

*Proposed.Construction.Type, Supervisor.District ) #从原数据集中挑选出选中的属性值构成新的待处理数据集；*

为了避免某些属性值空缺对关联规则挖掘带来的影响，先对所选择的属性中有缺失值的数据进行处理，在本实验中，选择直接删除含缺失值的数据，处理代码如下：

*data <- na.omit(data)*

为了便于关联规则挖掘，对属性的取值做了一定的处理，具体如下：

（1）Permit Type属性总共有8种类型，属性值分别为1~8，在关联规则挖掘时，为了便于区分，“1”用 “Permit type1” 表示，“2”用 “Permit type2” 表示，“3”用 “Permit type3” 表示，“4”用 “Permit type4” 表示，“5”用 “Permit type5” 表示，“6”用 “Permit type6” 表示，“7”用 “Permit type7” 表示，“8”用 “Permit type8” 表示；

（2）Street Number属性中，0~10用 “few streets” 表示,10~100用 “some streets” 表示, 100~1000用 “many streets” 表示，大于1000用 “plenty streets” 表示；

（3）Unit属性中，0~10用 “few units” 表示,10~100用 “some units” 表示, 100~1000用 “many units” 表示，大于1000用 “plenty units” 表示；

（4）Current.Status属性总共有14种类型，属性值描述保持不变；

（5） Number.of.Existing.Stories属性中，0~5用 “few existing stories” 表示,5~20用 “some existing stories” 表示, 20~50用 “many existing stories” 表示，大于50用 “plenty existing stories” 表示；

（6）Number.of.Proposed.Stories属性中，0~5用 “few proposed stories” 表示,5~20用 “some proposed stories” 表示, 20~50用 “many proposed stories” 表示，大于50用 “plenty proposed stories” 表示；

（7）Estimated.Cost属性中，0~1000用 “few revised cost” 表示,1000~10000用 “some revised cost” 表示, 10000~100000用 “many revised cost” 表示，大于100000用 “plenty revised cost” 表示；

（8）Revised.Cost属性中，0~1000用 “few estimates cost” 表示,1000~10000用 “some estimates cost” 表示, 10000~100000用 “many estimates cost” 表示，大于100000用 “plenty estimates cost” 表示；

（9）Existing.Units属性中，0~1000用 “few existing units” 表示,1000~10000用 “some existing units” 表示, 10000~100000用 “many existing units” 表示，大于100000用 “plenty existing units” 表示；

（10）Proposed.Units属性中，0~1000用 “few proposed units” 表示,1000~10000用 “some proposed units” 表示, 10000~100000用 “many proposed units” 表示，大于100000用 “plenty proposed units” 表示；

（11） Existing.Construction.Type属性，属性值分别为1~5，为了便于区分， “1” 用“existing construction type1”表示，“2” 用“existing construction type2”表示，“3” 用“existing construction type3”表示，“4” 用“existing construction type4”表示，“5” 用“existing construction type5”表示；

（12）Proposed.Construction.Type属性，属性值分别为1~5，为了便于区分， “1” 用“proposed construction type1”表示，“2” 用“proposed construction type2”表示，“3” 用“proposed construction type3”表示，“4” 用“proposed construction type4”表示，“5” 用“exi proposed sting construction type5”表示；

（13）Supervisor.District属性中，0~1000用 “few districts” 表示,1000~10000用 “some districts” 表示, 10000~100000用 “many districts” 表示，大于100000用 “plenty districts” 表示；

在R语言中的执行代码如下：

*#Permit Type属性总共有8种类型，分别为1~8， 保持不变*

*data[data$Permit.Type == '1', 'Permit.Type'] = 'Permit type1'*

*data[data$Permit.Type == '2', 'Permit.Type'] = 'Permit type2'*

*data[data$Permit.Type == '3', 'Permit.Type'] = 'Permit type3'*

*data[data$Permit.Type == '4', 'Permit.Type'] = 'Permit type4'*

*data[data$Permit.Type == '5', 'Permit.Type'] = 'Permit type5'*

*data[data$Permit.Type == '6', 'Permit.Type'] = 'Permit type6'*

*data[data$Permit.Type == '7', 'Permit.Type'] = 'Permit type7'*

*data[data$Permit.Type == '8', 'Permit.Type'] = 'Permit type8'*

*#Street Number属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Street.Number))*

*data[num >= 0 & num <10, 'Street.Number'] = 'few streets'*

*data[num >= 10 & num <100, 'Street.Number'] = 'some streets'*

*data[num >= 100 & num <1000, 'Street.Number'] = 'many streets'*

*data[num >= 1000, 'Street.Number'] = 'plenty streets'*

*#Unit属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Unit))*

*data[num >= 0 & num <10, 'Unit'] = 'few units'*

*data[num >= 10 & num <100, 'Unit'] = 'some units'*

*data[num >= 100 & num <1000, 'Unit'] = 'many units'*

*data[num >= 1000, 'Unit'] = 'plenty units'*

*#Current.Status属性总共有14种类型，保持不变*

*#Number.of.Existing.Stories属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Number.of.Existing.Stories))*

*data[num >= 0 & num <5, 'Number.of.Existing.Stories'] = 'few existing stories'*

*data[num >= 5 & num <20, 'Number.of.Existing.Stories'] = 'some existing stories'*

*data[num >= 20 & num <50, 'Number.of.Existing.Stories'] = 'many existing stories'*

*data[num >= 50, 'Number.of.Existing.Stories'] = 'plenty existing stories'*

*#Number.of.Proposed.Stories属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Number.of.Proposed.Stories))*

*data[num >= 0 & num <5, 'Number.of.Proposed.Stories'] = 'few proposed stories'*

*data[num >= 5 & num <20, 'Number.of.Proposed.Stories'] = 'some proposed stories'*

*data[num >= 20 & num <50, 'Number.of.Proposed.Stories'] = 'many proposed stories'*

*data[num >= 50, 'Number.of.Proposed.Stories'] = 'plenty proposed stories'*

*#Estimated.Cost属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Estimated.Cost))*

*data[num >= 0 & num <1000, 'Estimated.Cost'] = 'few estimates cost'*

*data[num >= 1000 & num <10000, 'Estimated.Cost'] = 'some estimates cost'*

*data[num >= 10000 & num <100000, 'Estimated.Cost'] = 'many estimates cost'*

*data[num >= 100000, 'Estimated.Cost'] = 'plenty estimates cost'*

*#Revised.Cost属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Revised.Cost))*

*data[num >= 0 & num <1000, 'Revised.Cost'] = 'few revised stories'*

*data[num >= 1000 & num <10000, 'Revised.Cost'] = 'some revised stories'*

*data[num >= 10000 & num <100000, 'Revised.Cost'] = 'many revised stories'*

*data[num >= 100000, 'Revised.Cost'] = 'plenty revised stories'*

*#Existing.Units属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Existing.Units))*

*data[num >= 0 & num <1000, 'Existing.Units'] = 'few existing units'*

*data[num >= 1000 & num <10000, 'Existing.Units'] = 'some existing units'*

*data[num >= 10000 & num <100000, 'Existing.Units'] = 'many existing units'*

*data[num >= 100000, 'Existing.Units'] = 'plenty existing units'*

*#Proposed.Units属性用few, some, many 和 plenty表示*

*num = as.numeric(as.character(data$Proposed.Units))*

*data[num >= 0 & num <1000, 'Proposed.Units'] = 'few proposed units'*

*data[num >= 1000 & num <10000, 'Proposed.Units'] = 'some proposed units'*

*data[num >= 10000 & num <100000, 'Proposed.Units'] = 'many proposed units'*

*data[num >= 100000, 'Proposed.Units'] = 'plenty proposed units'*

*#Existing.Construction.Type属性，用1~5表示*

*data[data$Existing.Construction.Type == '1', 'Existing.Construction.Type'] = 'existing construction type1'*

*data[data$Existing.Construction.Type == '2', 'Existing.Construction.Type'] = 'existing construction type2'*

*data[data$Existing.Construction.Type == '3', 'Existing.Construction.Type'] = 'existing construction type3'*

*data[data$Existing.Construction.Type == '4', 'Existing.Construction.Type'] = 'existing construction type4'*

*data[data$Existing.Construction.Type == '5', 'Existing.Construction.Type'] = 'existing construction type5'*

*#Proposed.Construction.Type属性，用1~5表示*

*data[data$Proposed.Construction.Type == '1', 'Proposed.Construction.Type'] = 'proposed construction type1'*

*data[data$Proposed.Construction.Type == '2', 'Proposed.Construction.Type'] = 'proposed construction type2'*

*data[data$Proposed.Construction.Type == '3', 'Proposed.Construction.Type'] = 'proposed construction type3'*

*data[data$Proposed.Construction.Type == '4', 'Proposed.Construction.Type'] = 'proposed construction type4'*

*data[data$Proposed.Construction.Type == '5', 'Proposed.Construction.Type'] = 'proposed construction type5'*

*#Supervisor.District属性，用few,some,many,plenty表示*

*num = as.numeric(as.character(data$Supervisor.District))*

*data[num >= 0 & num <1000, 'Supervisor.District'] = 'few districts'*

*data[num >= 1000 & num <10000, 'Supervisor.District'] = 'some districts'*

*data[num >= 10000 & num <100000, 'Supervisor.District'] = 'many districts'*

*data[num >= 100000, 'Supervisor.District'] = 'plenty districts'*

完成数据集的预处理之后，存储为.csv文件，用于关联规则的挖掘，执行代码为：

*write.csv(data, file = "preprocess\_Building\_Permits.csv", row.names = F)*

#### 3.2 关联规则挖掘

**找出频繁项集**

先将预处理得到的数据集读入，

*data = read.transactions("preprocess\_Building\_Permits.csv", format = "basket", sep = ",", rm.duplicates=TRUE)*

然后导入用于数据挖掘的包“arules”

*library(“arules”)*

R语言寻找频繁项集的代码如下：

*#得到频繁项集并保存*

*frequency = eclat(data, parameter = list(support = 0.05, maxlen = 6))*

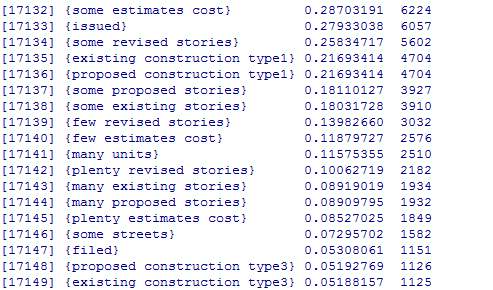
*write(frequency, file = "frequency.csv", sep = ",", quote = TRUE, row.names = FALSE)*

*#查看频繁项集*

*inspect(frequency)*

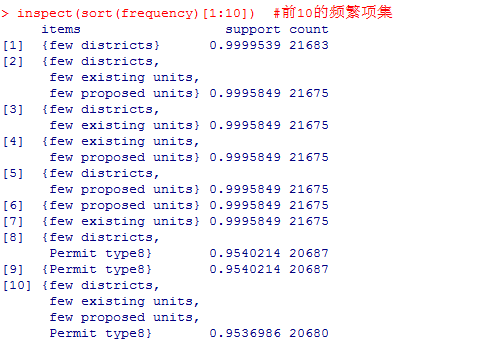
*inspect(sort(frequency)[1:10]) #前10的频繁项集*

得到的结果如下：



实验结果表明，符合约束条件的频繁项集有17149项。

查看频繁项集中出现频率最高的前10项，如下图：



从实验结果我们可以看出，几乎所有的建筑区中（>99.9%），每个监管人员只监管很少量的建筑（1~5），而且这些建筑中已存在单位和拟建立的单位数量都非常少（0~10）。

**导出关联规则**

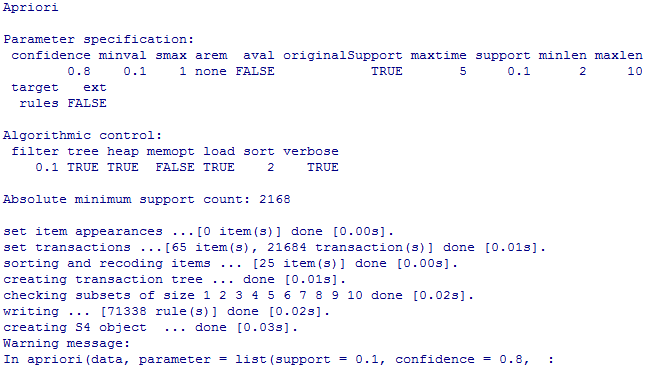
使用apriori算法进行关联规则挖掘，代码如下：

*#使用apriori算法得到关联规则*

*rules = apriori(data, parameter = list(support = 0.1, confidence = 0.8, minlen = 2))*

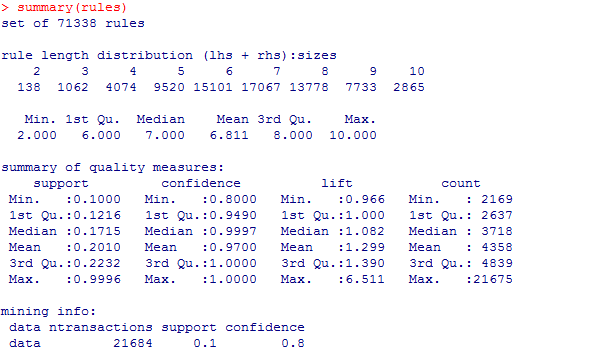
*summary(rules)*

结果如下图：



从返回的结果可以看出，总共有71338条规则生成。

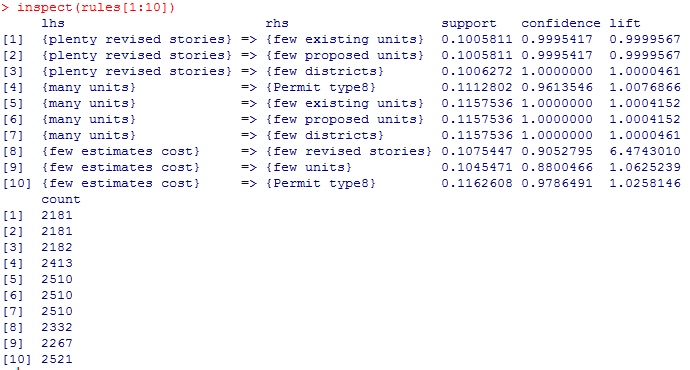
**查看规则的汇总信息**



由结果可知，长度为4的规则最多。

**查看具体关联规则**

*inspect(rules[1:10])*



返回的结果中已经包含了支持度、置信度等信息。

将得到的结果进行存储，执行：

*write(rules, file = "relation\_rule.csv", sep = ",", row.names = FALSE)*

**对规则进行评价**

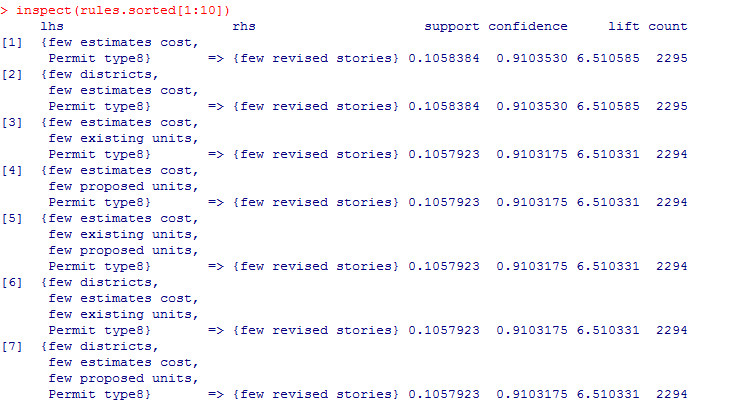
这里用Lift指标对规则进行评价，代码为：

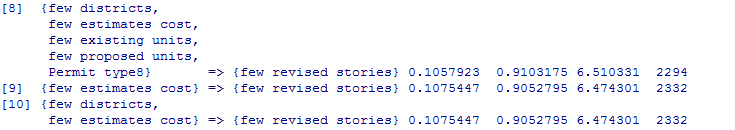
*#根据Lift对关联结果降序排序，并查看前10项*

*rules.sorted = sort(rules, decreasing = TRUE, by = "lift")*

*inspect(rules.sorted[1:10])*

返回的结果为：





返回的结果按照Lift值降序排列。从中可以看出比较有用的关联规则。如在第8种类型的许可下的建筑，当初始预算和修改后的工程预算较小时，相对应的故事（? Stories）也会对应较少。

4. 实验感悟

通过本次实验，进一步熟悉了用R语言进行数据挖掘。通过实验还了解到数据关联规则常用的算法apriori算法，并运用这种经典的算法从大量数据中挖掘出有用的关联规则。同时用Lift特征对挖掘的关联规则进行评价，实现了一个较完整的关联规则挖掘的一般过程。