## 数据探索性分析与数据预处理

余梦巧 2020171089

### 1. 问题描述

本次作业中，将对2个数据集进行探索性分析与预处理。

### 2. 数据说明

* 数据集1: NFL Play-by-Play 2009-2017
* 数据集2: San Francisco Building Permits

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

数据集中属性解释：

* 数据集1: [参考](https://github.com/ryurko/nflscrapR-data)
* 数据集2: 见下载地址中DataDictionaryBuildingPermit.xlsx

### 3. 数据分析要求

#### 3.1 数据可视化和摘要

##### 数据摘要

* 对标称属性，给出每个可能取值的频数，
* 数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数。

##### 数据的可视化

针对数值属性，

* 绘制直方图，用qq图检验其分布是否为正态分布。
* 绘制盒图，对离群值进行识别

#### 3.2 数据缺失的处理

观察数据集中缺失数据，分析其缺失的原因。

分别使用下列四种策略对缺失值进行处理:

* 将缺失部分剔除
* 用最高频率值来填补缺失值
* 通过属性的相关关系来填补缺失值
* 通过数据对象之间的相似性来填补缺失值

处理后，可视化地对比新旧数据集。

### 4. 实验过程

#### 4.1 San Francisco Building Permits数据集处理

本次实验总共需要对两个数据集进行处理，考虑到由简入繁的原则，先对相对比较小的第二个数据集（San Francisco Building Permits）进行处理。具体的实验过程如下：

**导入包**

为了完成本实验的相关操作，需要导入两个R语言的包，分别是：

library("DMwR")

library("car")

具体法人导入操作是，在控制台键入：

install.package("DMwR")

install.package("car")

进行安装。完成依赖包的导入之后，就可以进行相关的数据操作。

**设置路径**

将运行路径定位到项目所在的路径，针对于本机的这个项目，运行语句

setwd("C://Users//Meggie//Desktop//Building\_Permits")

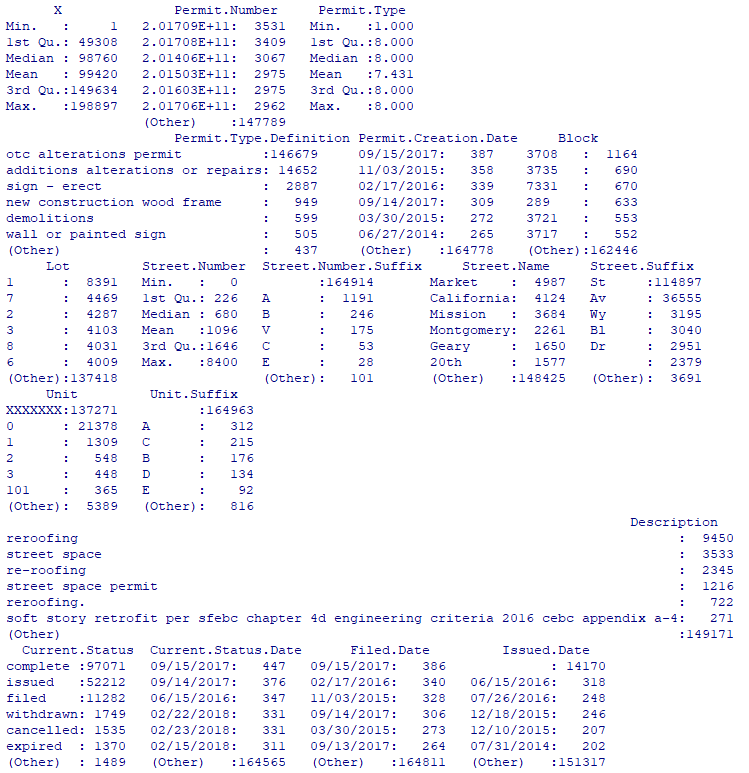
**导入数据**

本实验提供的数据为.csv格式，为了简便，直接用read.csv()函数读取，

data <- read.csv("Building\_Permits\_Similarity.csv", header = T, sep = ",")

**分析数据摘要**

在接触到一个数据处理任务之前，我们一般对这个领域缺少先验知识，因此，对数据有个概要了解是非常有必要的。概要信息包括数值属性的最大值、最小值、中值、均值、众数、四分位数。对于标称属性，一般指各值的出现频数和最高频数值。另外，对于属性的缺失值需要进行统计，这对我们分析数据剔除噪声数据、补全属性值等具有重要意义。以上数据摘要可以通过summary()函数得到，结果保存在项目文件夹下的.\Summary.txt中，部分数据摘要如下图所示：



由实验结果可以看出，数值属性返回了最大值、最小值、中位数、四分位数，而标称属性范围了频数最高的若干个属性值。这些值对于整体把握数据有非常重要的作用。

**数值属性可视化**

在做这一步之前，先要结合实验所给的属性解释判断每一个属性是否为数值属性，针对San Francisco Building Permits这个数据集的属性，经分析，确定为数值属性的为：Street.Number, Number.of.Existing.Stories,Number.of.Proposed.Stories, Estimated.Cost, Revised.Cost, Existing.Units 和Proposed.Units。接下来对这些属性绘制直方图，QQ图和盒图来可视化的对数据集进行分析。具体的会值代码以可视化“Street Number”这一属性为例。

jpeg(file=paste('Hist of','Street.Number', '.jpg' )) #绘制图片将被存储的路径

hist(data$Street.Number, prob = T) #绘制直方图

lines(density(data$Street.Number,na.rm=T)) #在直方图上绘制相应的曲线

rug(jitter(data$Street.Number)) #在X轴附近绘制变量的实际值，从而容易识别离群点

dev.off( )

jpeg(file=paste('QQ of ','Street.Number','.jpg') )

qqPlot(data$Street.Number,main=paste('Normal QQ plot of ','Street.Number'),ylab= as.character('Street.Number')) #绘制QQ图，检验属性值是否符合正态分布

dev.off( )

jpeg(file=paste('box of ','Street.Number','.jpg') )

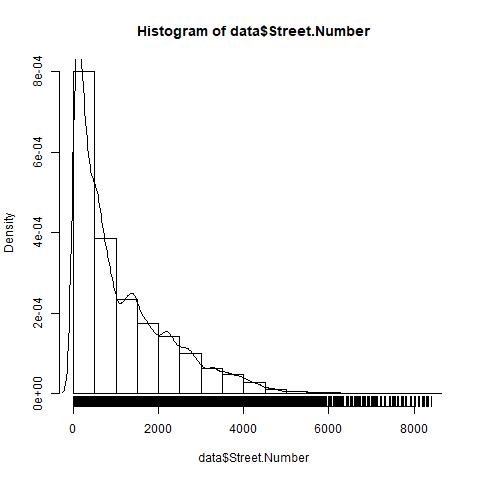
boxplot(data$Street.Number,ylab= as.character('Street.Number')) #绘制盒图，对离群值进行识别

rug(jitter(data$Street.Number),side=2)

abline(h=mean(data$Street.Number,na.rm=T),lty=2)

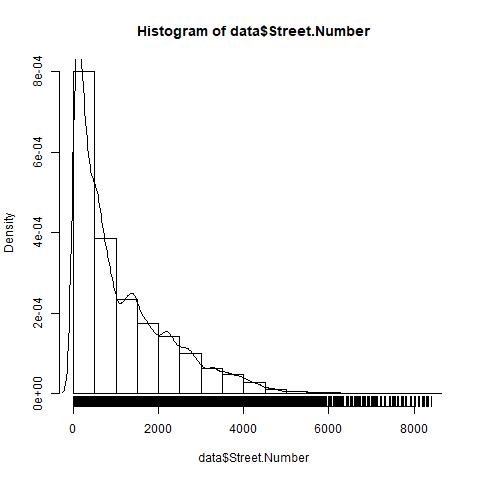
dev.off( )

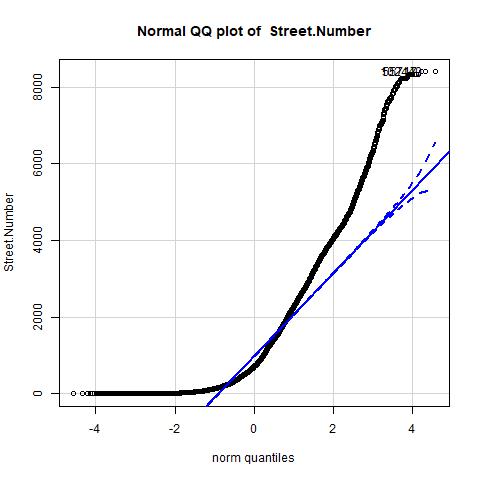
下图是属性“Street Number”所绘制得到的直方图



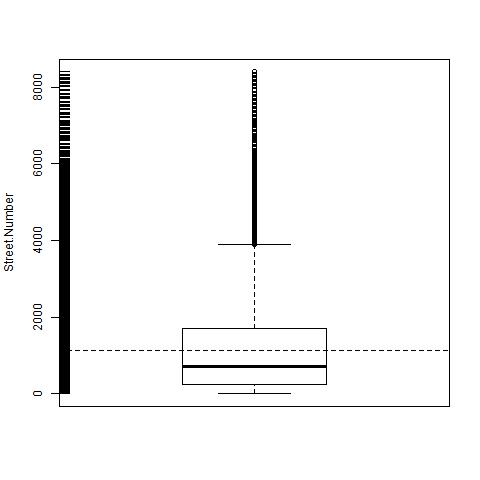
通过直方图。可以发现大部分的属性值落在0~2000之间

下图右图是用qqPlot()得到的QQ图它绘制变量值和正态分布的理论分位数(黑线)的散点图。同时，它给出正态分布的 95%置信区间的带状图（蓝色虚线）。从下图可知，0~2000的之间的属性值接近于正态分布，剩余属性值偏离正态分布较远。





接下是盒图，对离群值进行识别



图中，水平盒子的上下边界代表数据的两个四分位点Q1、Q3，Q1与Q3的差值为IQR，即IQR=Q3-Q1，盒子之外的两条水平直线代表观测值的最小值和最大值，取值分别是Q1-1.5\*IQR和Q3+1.5\*IQR，粗体的水平线为中位数，圆圈代表着离散点。此外，我们也添加了一条水平虚线，代表着均值，数据的实际分布在右侧标上，作为对应。通过此图可以更加直观地发现离散点的分布状况。

本实验中其他数值属性的可视化效果图在本项目路径下的.\结果图\Original\下。

**数据缺失的处理**

这里总共用四种方法对确实的数据进行处理，分别是

1. 将缺失部分剔除
2. 用最高频值来填补缺失值
3. 通过属性的相关性来填补
4. 通过数据对象之间的相似性来填补缺失值

第一种方法需执行：

data\_delete <- na.omit(data)

这种方法最为直接简单，直接去掉有缺失属性的数据，这种方法适合具有少量缺失的情况，直接去掉之前一般需要对数据进行一次检查。此方法的缺点是，一旦缺失属性的数据多了，直接去掉将对后面的数据分析和数据挖掘产生不好作用。

第二种方法需执行：

data\_most <- centralImputation(data)

顾名思义，这种方法通过最高频率值来替换缺失值，此外也可以通过使用中位值、均值等来进行替换。

第三种方法利用属性变量之间的相关性进行补全。首先去除缺省情况太多的数据，然后求出所有数值属性之间的相关性矩阵，本次实验中，经验证发现“Number of Existing Stories”与“Number of Proposed Stories”具有较强相关性。再在这两个模型之间建立回归模型，例如这里两个变量，选择的是线性回归模型，得到最优模型参数，通过该模型值为缺失的这两个属性进行补全。如下图所示，我们可以得到两者之间的关系为：Number.of.Existing.Stories = -0.0171 + 0.9968 \* Number.of.Proposed.Stories。以这两个属性的相关性为例，执行代码为：

symnum(cor(data[,21:22], use="complete.obs"))

lm(Number.of.Existing.Stories ~ Number.of.Proposed.Stories , data = data)

fillNumber.of.Existing.Stories <- function(Number.of.Proposed.Stories){

if(is.na(Number.of.Proposed.Stories))

return(NA)

else return (-0.0171 + 0.9968 \* Number.of.Proposed.Stories)

}

data\_cor[is.na(data\_cor$Number.of.Existing.Stories),'Number.of.Existing.Stories']<-

sapply(data\_cor[is.na(data\_cor$Number.of.Existing.Stories),'Number.of.Proposed.Stories'],fillNumber.of.Existing.Stories)

第四种方法利用数据对象之间的相似性进行补全，在数据集中寻找最为相似的数据，按照它的值来补全缺失值。这个思路扩展一下，也可以采用寻找最为接近的k个数据，按照一定权值进行组合。在本实验中k值去10，其执行代码为：

data\_similar <- data[-manyNAs(data),]

data\_similar = knnImputation(data,k=10)

用这四种方式对数据集中的缺失数据进行处理之后，都需要对处理后的数据进行存储，分别执行：

write.csv(data,file = "Building\_Permits\_Correalation.csv")

即可。对于确实处理之后得到的数据，再进行属性的可视化，分别绘制直方图、QQ图和盒图，和之前的数据进行可视化对比。数据集的可视化帮助我们更直观的了解数据的分布。

至此，就完成了对San Francisco Building Permits数据集的处理。对于NFL Play-by-Play 2009-2017的处理方法类似，下面进行简述。

#### 4.2 NFL Play-by-Play 2009-2017数据集处理

**导入包**

与处理数据集1类似，为了完成实验的相关操作，需要导入两个R语言的包，分别是：

library("DMwR")

library("car")

**设置路径**

将运行路径定位到项目所在的路径，针对于本机的这个项目，运行语句

setwd("C://Users//Meggie//Desktop// NFL Play by Play")

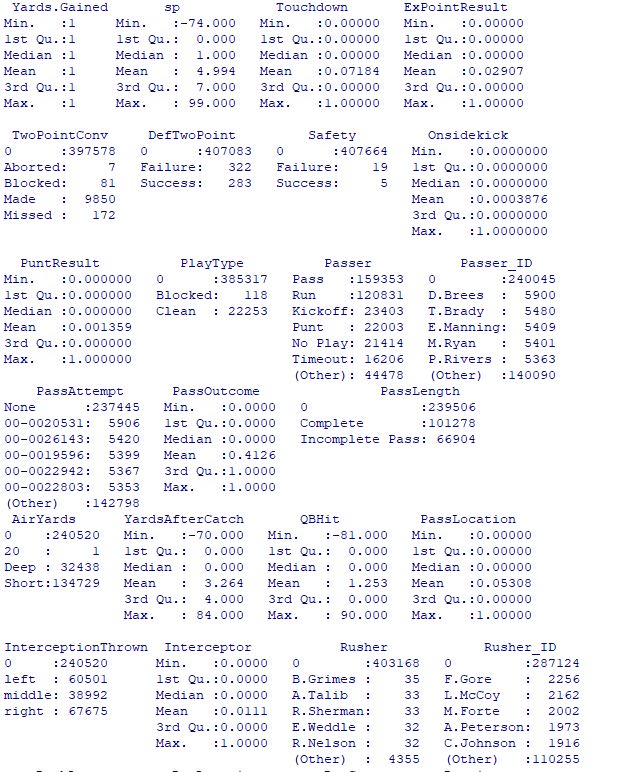
**导入数据**

本实验提供的数据为.csv格式，为了简便，直接用read.csv()函数读取，

data <- read.csv("NFL Play by Play 2009-2017 (v4) .csv", header = T, sep = ",")

**分析数据摘要**

以上数据摘要可以通过summary()函数得到，结果保存在项目文件夹下的.\Summary.txt中，部分数据摘要如下图所示：



**数值属性可视化**

在做这一步之前，先要结合实验所给的属性解释判断每一个属性是否为数值属性，针对NFL Play-by-Play 2009-2017这个数据集的属性，经分析，确定为数值属性的为：'Date','GameID','Drive','qtr','down','time','TimeUnder','TimeSecs','PlayTimeDiff','SideofField','yrdln','yrdline100','ydstogo','ydsnet','GoalToGo','FirstDown','posteam','DefensiveTeam','desc','PlayAttempted','Yards.Gained','sp','Touchdown','ExPointResult','TwoPointConv','DefTwoPoint','Safety','Onsidekick','PuntResult','PlayType','Passer','Passer\_ID','PassAttempt','PassOutcome','PassLength','AirYards','YardsAfterCatch','QBHit','PassLocation','InterceptionThrown','Interceptor','Rusher','Rusher\_ID','RushAttempt','RunLocation','RunGap','Receiver','Receiver\_ID','Reception','ReturnResult','Returner','BlockingPlayer','Tackler1','Tackler2','FieldGoalResult','FieldGoalDistance','Fumble','RecFumbTeam','RecFumbPlayer','Sack','Challenge.Replay','ChalReplayResult','Accepted.Penalty','PenalizedTeam','PenaltyType','PenalizedPlayer','Penalty.Yards','PosTeamScore','DefTeamScore','ScoreDiff','AbsScoreDiff','HomeTeam','AwayTeam','Timeout\_Indicator','Timeout\_Team','posteam\_timeouts\_pre','HomeTimeouts\_Remaining\_Pre','AwayTimeouts\_Remaining\_Pre','HomeTimeouts\_Remaining\_Post','AwayTimeouts\_Remaining\_Post','No\_Score\_Prob','Opp\_Field\_Goal\_Prob','Opp\_Safety\_Prob','Opp\_Touchdown\_Prob','Field\_Goal\_Prob','Safety\_Prob','Touchdown\_Prob','ExPoint\_Prob','TwoPoint\_Prob','ExpPts','EPA','airEPA',' yacEPA','Home\_WP\_pre','Away\_WP\_pre','Home\_WP\_post','Away\_WP\_post','Win\_Prob','WPA','airWPA','yacWPA'和'Season'。接下来对这些属性绘制直方图，QQ图和盒图来可视化的对数据集进行分析。因为这组数据的数值属性过多，用穷举的方式已经不合适，因此采用循环结构：

for (i in 1:102)

{

a <- data[,i] #第i个属性的全部取值

attr <- colnames(data[i]) # 得到第i个属性的属性名称

if (attr %in% value\_attr) #当第i个属性为数值属性时，执行可视化的画图操作

{

jpeg(file=paste('Histogram of ',attr,'.jpg') )

hist(a, prob=T, xlab='',main=paste('Histogram of ',attr),ylim=0:1)

lines(density(a,na.rm=T))

rug(jitter(a))

dev.off( )

jpeg(file=paste('QQ of ',attr,'.jpg') )

qqPlot(a,main=paste('Normal QQ plot of ',attr),ylab= as.character(attr))

dev.off( )

jpeg(file=paste('box of ',attr,'.jpg') )

boxplot(a,ylab= as.character('GameID'))

rug(jitter(a),side=2)

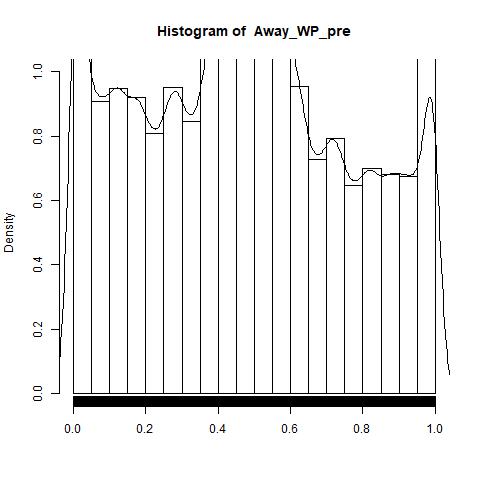
abline(h=mean(a,na.rm=T),lty=2)

dev.off( )

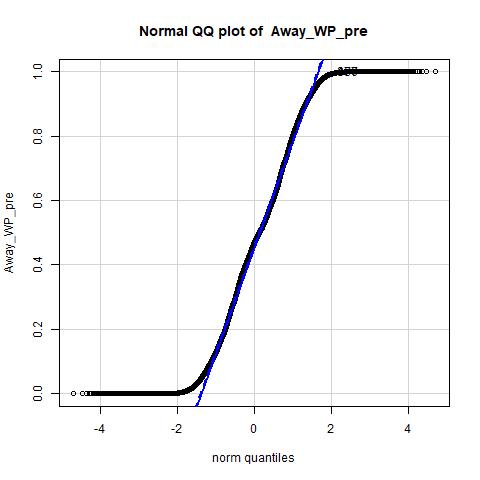
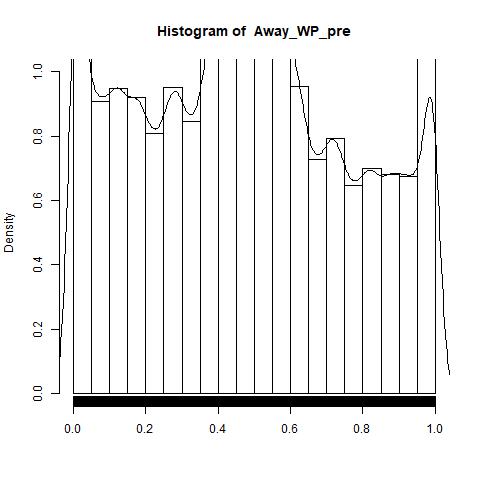
}

}

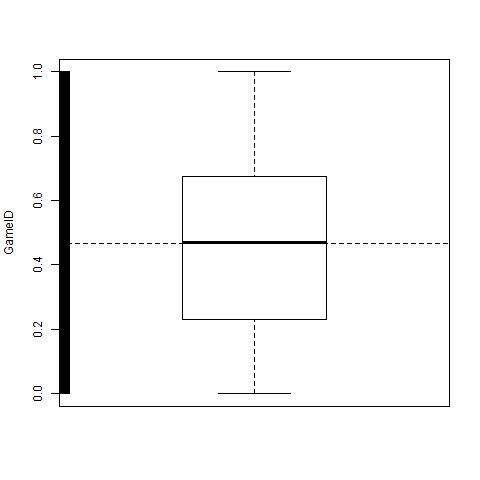
下图是属性“Away\_WP\_pre”所绘制得到的直方图



下图右图是用qqPlot()得到的QQ图它绘制变量值和正态分布的理论分位数(黑线)的散点图。从下图可知，在-1.7~1.7的之间”Away WP pre”的Q-Q拟合线趋近于直线，说明改属性值在这一区间内接近正态分布，剩余属性值则偏离正态分布较远。



接下是盒图，对离群值进行识别

****

本实验中其他数值属性的可视化效果图在本项目路径下的.\结果图\Original\下。

**数据缺失的处理**

这里总共用四种方法对确实的数据进行处理，分别是

1. 将缺失部分剔除
2. 用最高频值来填补缺失值
3. 通过属性的相关性来填补
4. 通过数据对象之间的相似性来填补缺失值

这里1、2、4种方法处理缺失值的操作与处理第一个数据集完全一致，不再赘述。这里着重分析第三种方法对于缺失值的处理。以“Home\_WP\_post”和“Away\_WP\_post”为例，经验证，这两个属性有较强的相关性，在这两个属性之间建立线性回归模型，得到Home\_WP\_post = 0.9992 + -0.9973\* Away\_WP\_post，于是缺失处理的代码可以写成：

symnum(cor(data[,76:102], use="complete.obs"))

fillHome\_WP\_post <- function(Away\_WP\_post){

if(is.na(Away\_WP\_post))

return(NA)

else return (-0.0171 + 0.9968 \* Away\_WP\_post)

}

data\_cor[is.na(data\_cor$Home\_WP\_post),'Home\_WP\_post']<-

sapply(data\_cor[is.na(data\_cor$Home\_WP\_post),'Away\_WP\_post'],fillHome\_WP\_post)

用这四种方式对数据集中的缺失数据进行处理之后，都需要对处理后的数据进行存储，分别执行：

write.csv(data,file = "Building\_Permits\_Correalation.csv")

即可。对于确实处理之后得到的数据，再进行属性的可视化，分别绘制直方图、QQ图和盒图，和之前的数据进行可视化对比。

至此，就完成了对San Francisco Building Permits数据集的处理。

### 实验收获

通过本次实验，认识到在大数据时代，数据量的庞大与复杂，通过人眼直观地观察已经完全无法解决问题，因此运用程序的手段对大数据进行分析尤为重要。其次，海量的数据中，部分数据缺失的现象非常普遍，如何处理缺失数据对于有效地利用数据也有至关重要的作用。本次实验学到了数据预处理和分析的基本方法和步骤，并对运用R语言进行数据预处理和分析有了一定的认知。