数据挖掘作业一

——数据探索性分析与数据预处理

姓名： 高建花

班级： 硕士 4 班

学号： 2120171010

数据探索性分析与数据预处理

1. 问题描述

对数据集 1：NFL Play-by-Play 2009-2017 和数据集 2：San Francisco Building Permits 进行探索性分析与预处理

2. 实验环境

Item	Description
Language	R
IDE	RGui
Package	pandas; base；car；DMwR

3. 数据分析要求

3.1 数据可视化和摘要（数据集 1）

3.1.1 数据摘要

先利用 R 语言中的 read.csv 读取数据，然后通过 summary 函数来得到数据的统计信息。核心代码如下：

```
#读取数据
data_path = paste(base_path, "作业1数据集/", sep="")
setwd(data_path)
data <- read.csv("NFL Play by Play 2009-2017 (v4).csv")

#数据摘要
summary(data)
```

数据集部分属性的统计信息分别如下图所示，可用类似的方法得到任何一个属性的统计信息。

对于标称属性，比如 Date、time、SideofField，summary 函数给出了每个可能取值的频数，考虑到数据量较大，这里只列出了频数最高的前六个取值；对于数值属性，比如 GameID、Drive、qtr、down 等，summary 函数给出了最小值（Min）、第一个四分位数（1st Qu）、中位数（Median）、均值（mean）、第三个四分位数（3rd Qu）、最大值（Max）、缺失值(NA's)。

```
      锘緿ate          GameID             Drive            qtr             down            time
2016-01-03:  2872   Min.   :2.009e+09   Min.   : 1.00   Min.   :1.000   Min.   :1       15:00  : 11792
2012-01-01:  2825   1st Qu.:2.011e+09   1st Qu.: 6.00   1st Qu.:2.000   1st Qu.:1       02:00  :  7012
2017-01-01:  2819   Median :2.013e+09   Median :12.00   Median :3.000   Median :2       00:00  :  6914
2017-12-31:  2801   Mean   :2.013e+09   Mean   :12.32   Mean   :2.577   Mean   :2       14:55  :  1174
2011-01-02:  2772   3rd Qu.:2.015e+09   3rd Qu.:18.00   3rd Qu.:4.000   3rd Qu.:3       01:55  :  1001
2014-12-28:  2771   Max.   :2.017e+09   Max.   :35.00   Max.   :5.000   Max.   :4       14:54  :   980
(Other)   :390828                                                       NA's   :61154   (Other):378815
  TimeUnder         TimeSecs      PlayTimeDiff      SideofField
Min.   : 0.000   Min.   :-900   Min.   :  0.00   OAK    : 13270
1st Qu.: 3.000   1st Qu.: 778   1st Qu.:  5.00   CLE    : 13212
Median : 7.000   Median :1800   Median : 17.00   BUF    : 13094
Mean   : 7.374   Mean   :1695   Mean   : 20.58   TEN    : 12968
3rd Qu.:11.000   3rd Qu.:2585   3rd Qu.: 37.00   MIA    : 12924
Max.   :15.000   Max.   :3600   Max.   :943.00   (Other):341692
                 NA's   :224    NA's   :444      NA's   :   528
```

注 1: 第一项为 Date，显示出来有点乱码。

3.1.2 数据的可视化

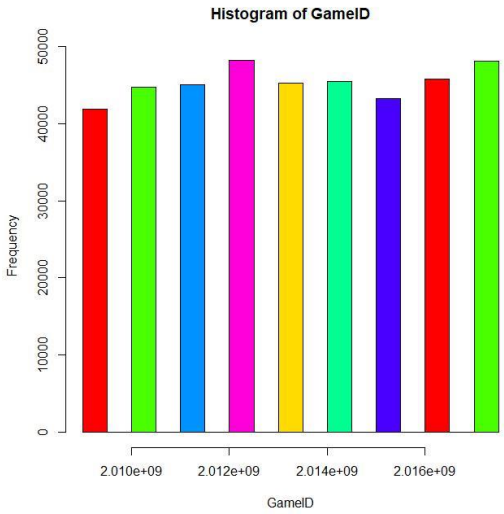
针对数值属性，分别绘制直方图、qq 图和盒图，核心代码如下。

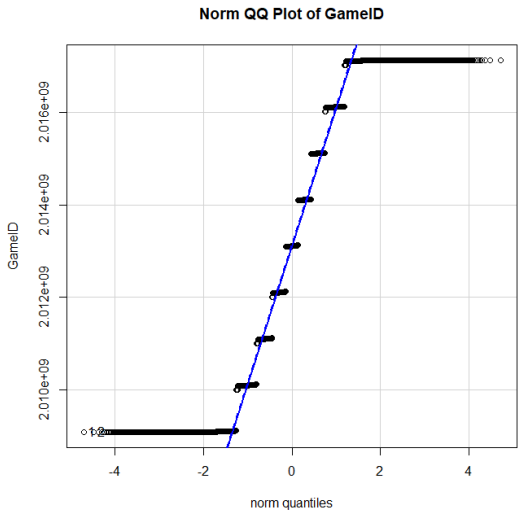
```
#绘制直方图
setwd(base_path)
dir.create("hist")
hist_path = paste(base_path, "hist/", sep="")
setwd(hist_path)

for (i in 2:ncol(data))
    if (class(data[, i]) != "factor")
    {
        hist(data[[i]], col=rainbow(7), xlab=names(data[i]), main=paste("Histogram of", names(data[i])))
        savePlot(paste("hist_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

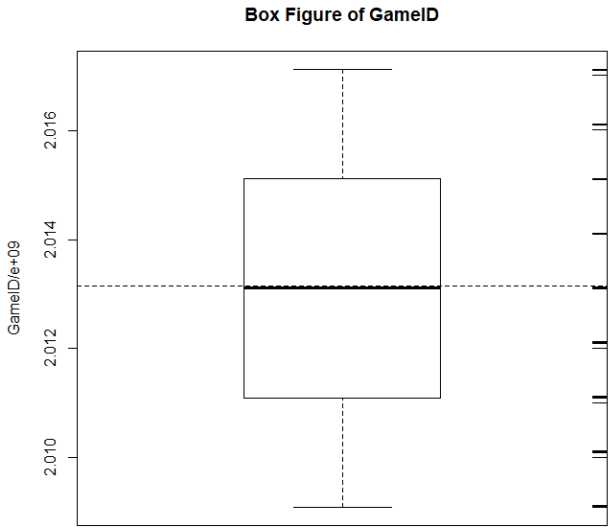
```
#绘制QQ图
setwd(base_path)
dir.create("qqFigure")
qqPlot_path = paste(base_path, "qqFigure/", sep="")
setwd(qqPlot_path)

for (i in 2:ncol(data))
    if (class(data[, i]) != "factor")
    {
        qqPlot(data[[i]], col=rainbow(7), main=paste('Norm QQ Plot of', names(data[i])), ylab=names(data[i]))
        savePlot(paste("qqPlot_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

```
#绘制盒图
setwd(base_path)
dir.create("boxPlot")
boxPlot_path = paste(base_path, "boxPlot/", sep="")
setwd(boxPlot_path)

#GameID数值较大，特殊处理
boxplot(data$GameID/1000000000, main="Box Figure of GameID", ylab="GameID/e+09")
rug(data$GameID/1e+09, side=4)
abline(h=mean(data$GameID/1e+09, na.rm=T), lty=2)
savePlot("boxPlot_GameID", type=c("jpg"))

for (i in 3:ncol(data))
    if (class(data[, i]) != "factor")
    {
        boxplot(data[[i]], main=paste('Box Figure of', names(data[i])), ylab=names(data[i]))
        rug(data[[i]], side=4)
        abline(h=mean(data[[i]], na.rm=T), lty=2)
        savePlot(paste("boxPlot_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

以 GameID 属性为例，其直方图、qq 图和盒图分别如下所示。







3.2 数据缺失的处理（数据集 1）

3.2.1 将缺失部分剔除

通过观察数据发现，数据集中某些属性的缺失值较多，当这些属性非空时，对应的同一行的其他列恰好存在空项，故以一行中只要有缺失数据就剔除这行的标准来处理数据集的话，所有的数据项都将会被剔除。所以此种缺失数据的处理方法并不适合此数据集。直接使用na.omit()函数来剔除掉缺失值。

剔除缺失部分的代码如下：

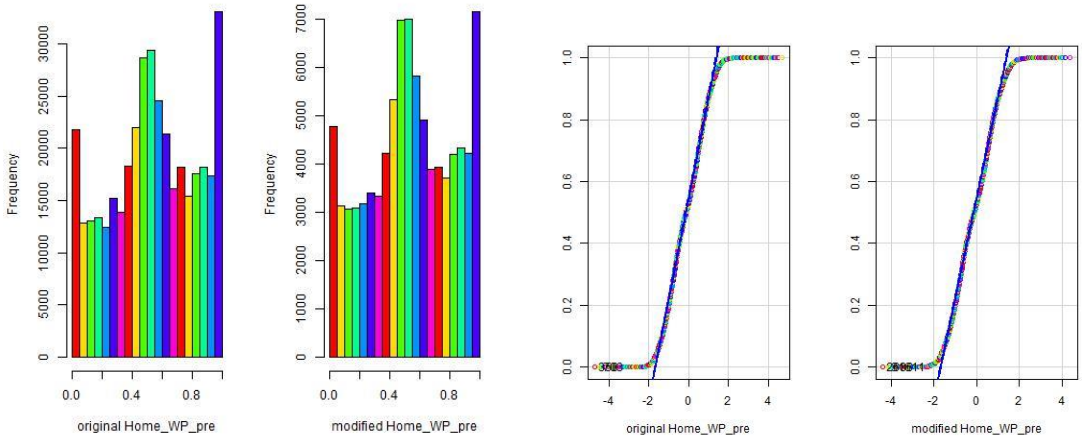
```
#将缺失部分剔除
data_omit1 = na.omit(data_omit1)
```

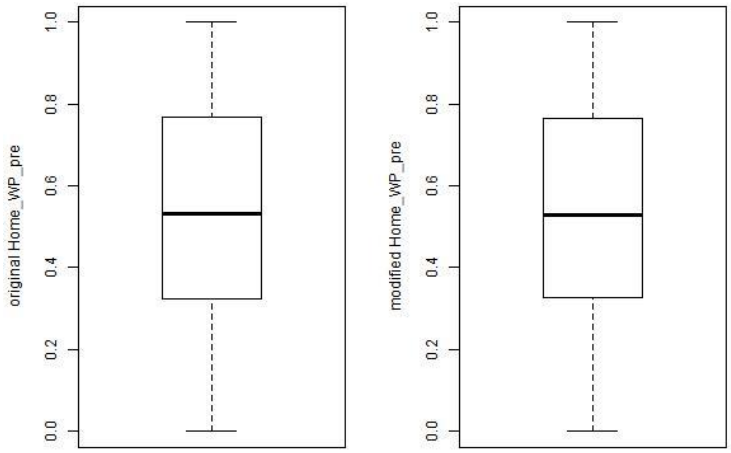
3.2.2 用最高频率来填补缺失值

对于数值属性的数据，用其中位数来填补缺失值；对于标签属性的数据，用其众数来填补缺失值。首先利用 manyNAs(data, nORp)函数来查找数据框 data 中缺失值过多（≥缺失比例 nORp）的行，nORp 默认为 0.2，即缺失值个数≥列数的 20%。然后利用 DMwR 包中的 centralImputation 函数来填补数据。核心代码如下：

```
#用最高频率来填补缺失值
data_omit2 = data[-manyNAs(data),]
data_omit2 = centralImputation(data_omit2)
```

以 Home_WP_pre 属性为例，分别对比填补缺失值前后的直方图、qq 图、盒图





对比发现，填补前后两者的直方图、qq 图和盒图变化都很小，所以此种填补缺失值的方法较为靠谱。

3.2.3 通过属性的相关关系来填补缺失值

先探索变量之间的相关关系，找到相关性较高的两个变量后，在寻找他们之间的线性回

归关系，最后通过线性回归关系计算缺失值进行填补。

先通过 cor()函数来产生变量之间的相关值矩阵，设定参数 use=”complete.obs”可以使 R 在计算相关值时忽略含有 NA 值的样本，然后用 symnum()函数来改善结果的输出形式，得到结果如下：

```
> symnum(cor(data[83:102],use="complete.obs"))
                O_S O_T F S_ T_ EP_ TP ExP EPA aE yE Hm_WP_pr Awy_WP_pr Hm_WP_ps Awy_WP_ps W_ WP aW yW Ss
Opp_Safety_Prob 1
Opp_Touchdown_Prob ,  1
Field_Goal_Prob .  .  1
Safety_Prob     ,     1
Touchdown_Prob  .     .  1
ExPoint_Prob                1
TwoPoint_Prob                  1
ExpPts          ,  ,  ,  ,        1
EPA                                  1
airEPA                                  1
yacEPA                               ,  .  1
Home_WP_pre                                 1
Away_WP_pre                                 1        1
Home_WP_post                                B        B        1
Away_WP_post                                B        B        1        1
Win_Prob                                                                    1
WP                                   +      .                                   1
airWP                                   ,  .                                        1
yacWP                                .  .  ,                                        ,  ,  1
Season                                                                                    1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

拿 Home_WP_pre 和 Away_WP_pre 举例来说，它们之间的相关度为 1。用 lm 函数来得到两者之间的相关关系如下。然后利用此回归关系填补 Home_WP_pre 属性的缺失值。

```
> lm(formula = Home_WP_pre~Away_WP_pre, data = data)

Call:
lm(formula = Home_WP_pre ~ Away_WP_pre, data = data)

Coefficients:
(Intercept)  Away_WP_pre
     0.9991      -0.9970
```

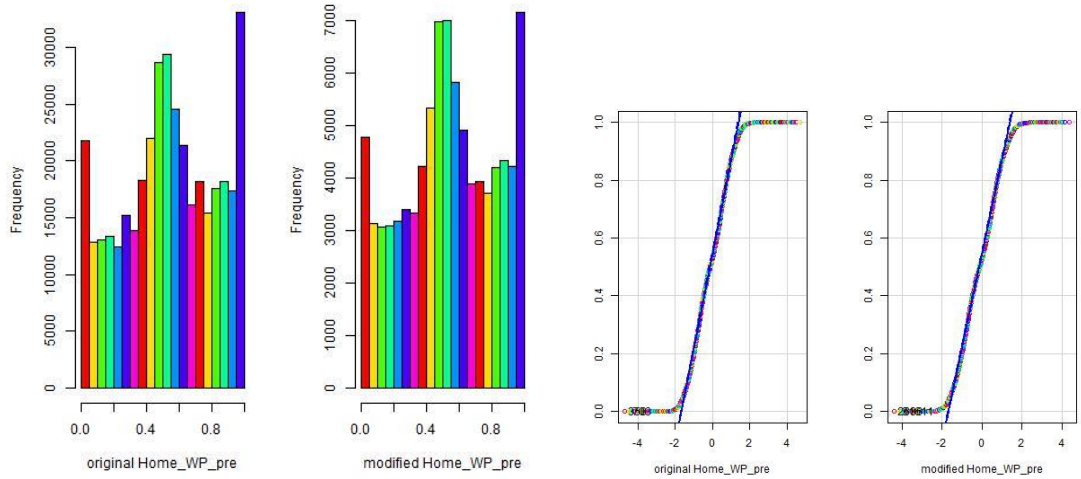
此部分的核心代码如下：

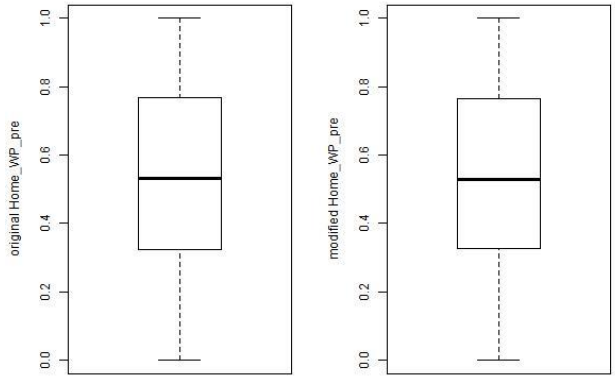
```
#通过属性的相关关系来填补缺失值
symnum(cor(data[83:102],use="complete.obs"))
lm(formula = Home_WP_pre~Away_WP_pre, data = data)
data_omit3 = data[-manyNAs(data),]
fillHome_WP_pre <- function(Away_WP_pre){
    if(is.na(Away_WP_pre))
        return(NA)
    else
        return (0.9991-0.9970*Away_WP_pre)
}
data_omit3[is.na(data_omit3$Home_WP_pre), 'Home_WP_pre'] <-
    sapply(data_omit3[is.na(data_omit3$Home_WP_pre),'Away_WP_pre'],fillHome_WP_pre)
```

对比两者的统计信息：

```
> summary(data$Home_WP_pre)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  0.000   0.325   0.531   0.534   0.769   1.000   24954
> summary(data_omit3$Home_WP_pre)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.3273  0.5292  0.5329  0.7655  1.0000
```

对比两者的直方图、qq 图和盒图：





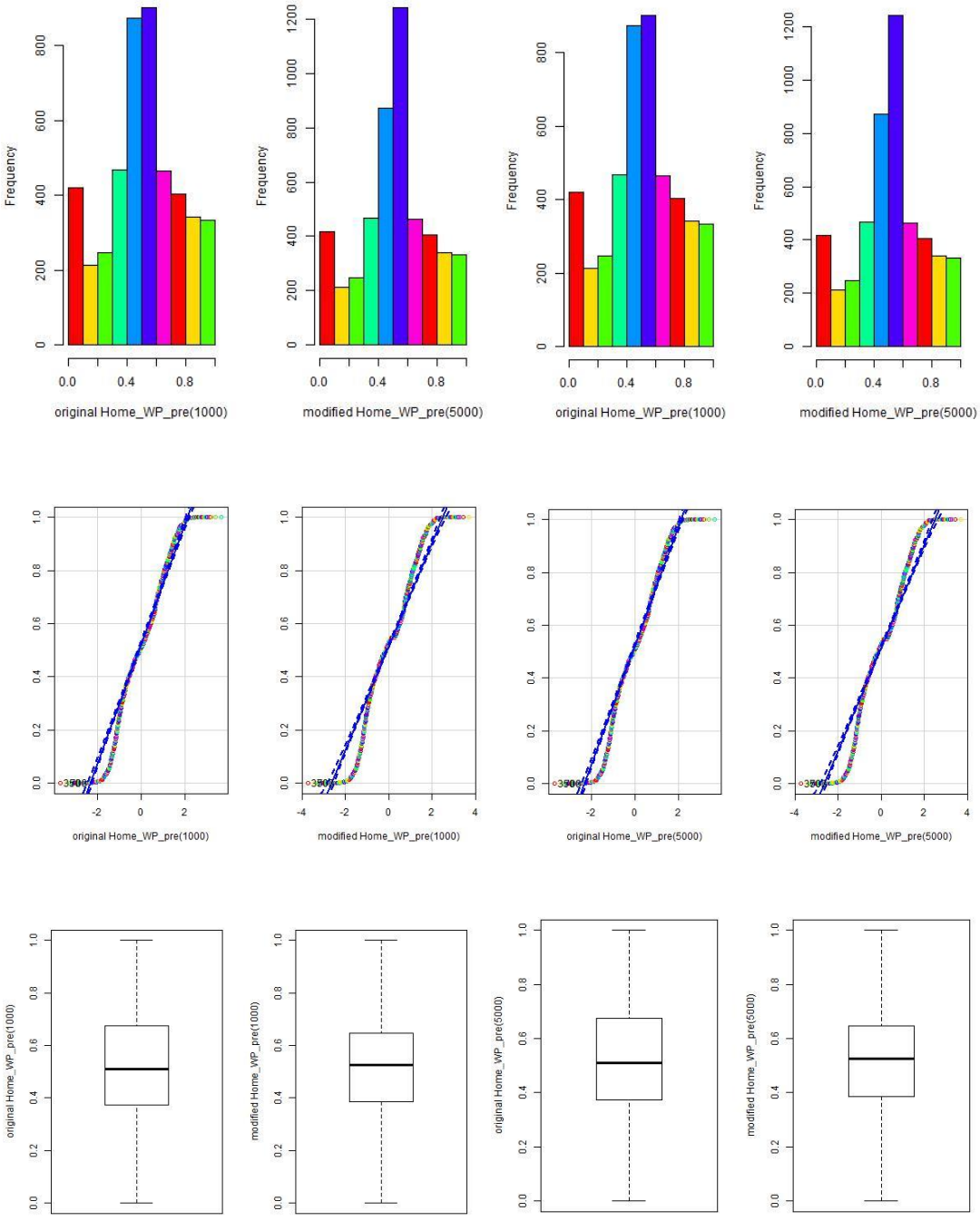
对比发现，填补前后两者的直方图、qq 图和盒图变化都很小，所以此种填补缺失值的方法较为靠谱。

3.2.4 通过数据对象之间的相似性来填补缺失值

可利用 KNN-均值来填充缺失值，其核心代码如下:

```
#通过数据对象之间的相似性来填补缺失值
data_1 = data[1:1000, 94:95]
data_2 = data[1:5000, 94:95]
data_omit4_1 = knnImputation(data_1, 10)
data_omit4_2 = knnImputation(data_2, 10)
```

考虑到数据集太大，此处只对前 1000 行和前 5000 行，其中两列数据进行了处理。以 Home_WP_pre 为例，其直方图、qq 图和盒图对比如下，左侧对 1000 行数据进行缺失处理的结果，右侧为对 5000 行数据进行缺失处理的结果。



相比较而言，较前三种处理方式更准确合理

4. 数据分析要求

4.1 数据可视化和摘要（数据集 2）

4.1.1 数据摘要

先利用 R 语言中的 read.csv 读取数据，然后通过 summary 函数来得到数据的统计信息。核心代码如下：

```
#读取数据
data_path = paste(base_path, "作业1数据集/", sep="")
setwd(data_path)
data <- read.csv("Building_Permits.csv")

#数据摘要
summary(data)
```

数据集部分属性的统计信息分别如下图所示，可用类似的方法得到任何一个属性的统计信息。

对于标称属性，比如 Permit.Number、Permit.Type.Definition 等，summary 函数给出了每个可能取值的频数，考虑到数据量较大，这里只列出了频数最高的前六个取值；对于数值属性，比如 Street.Number、Permit.Type 等，summary 函数给出了最小值（Min）、第一个四分位数（1st Qu）、中位数（Median）、均值（mean）、第三个四分位数（3rd Qu）、最大值（Max）、缺失值(NA's，此两项正好没有缺失值)。

```
     Permit.Number     Permit.Type
201602179765:   101   Min.   :1.000
201602179758:    66   1st Qu.:8.000
201602179775:    30   Median :8.000
201409166451:     9   Mean   :7.522
201702239990:     9   3rd Qu.:8.000
201708165004:     9   Max.   :8.000
(Other)     :198676
                    Permit.Type.Definition Permit.Creation.Date
otc alterations permit          :178844     09/15/2017:   413
additions alterations or repairs: 14663     11/03/2015:   396
sign - erect                    :  2892     02/17/2016:   363
new construction wood frame     :   950     09/14/2017:   335
demolitions                     :   600     06/27/2014:   307
wall or painted sign            :   511     03/30/2015:   298
(Other)                         :   440     (Other)   :196788
    Block            Lot         Street.Number  Street.Number.Suffix
3708   :  1195   001    : 10114   Min.   :   0          :196684
3735   :   750   007    :  5317   1st Qu.: 235   A      :  1501
7331   :   680   002    :  5183   Median : 710   B      :   291
0289   :   640   003    :  5042   Mean   :1122   V      :   228
3709   :   584   006    :  4835   3rd Qu.:1700   C      :    56
3717   :   578   008    :  4773   Max.   :8400   E      :    28
(Other):194473   (Other):163636                  (Other):   112
    Street.Name     Street.Suffix
Market    :  5443   St     :138358
California:  4587   Av     : 43219
Mission   :  4209   Bl     :  3555
Montgomery:  2403   Wy     :  3540
Geary     :  1966   Dr     :  3267
20th      :  1859          :  2768
(Other)   :178433   (Other):  4193
```

4.1.2 数据的可视化

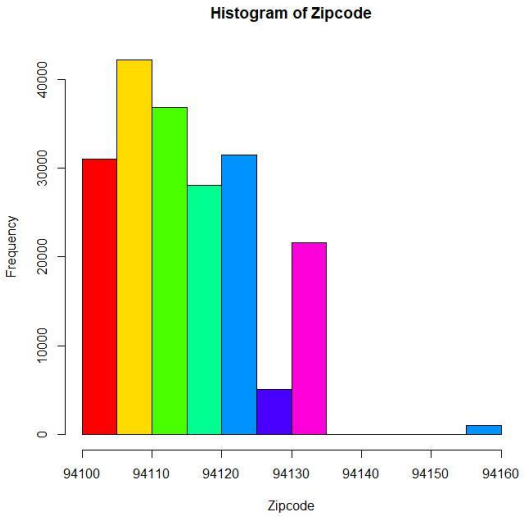
针对数值属性，分别绘制直方图、qq 图和盒图，核心代码如下。

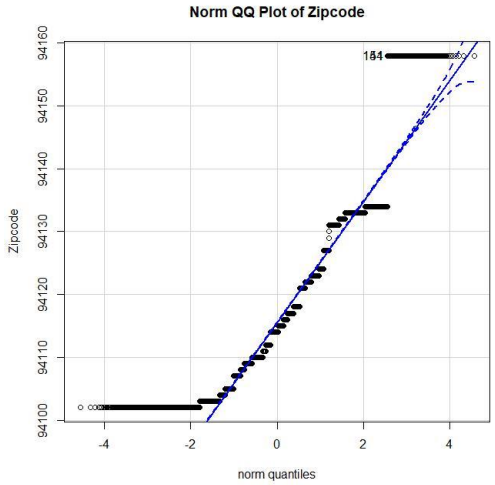
```r
#绘制直方图
setwd(base_path)
dir.create("hist")
hist_path = paste(base_path, "hist/", sep="")
setwd(hist_path)

for (i in 1:ncol(data))
    if (class(data[, i]) != "factor")
    {
        hist(data[[i]], col=rainbow(7), xlab=names(data[i]),
            main=paste("Histogram of", names(data[i])))
        savePlot(paste("hist_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

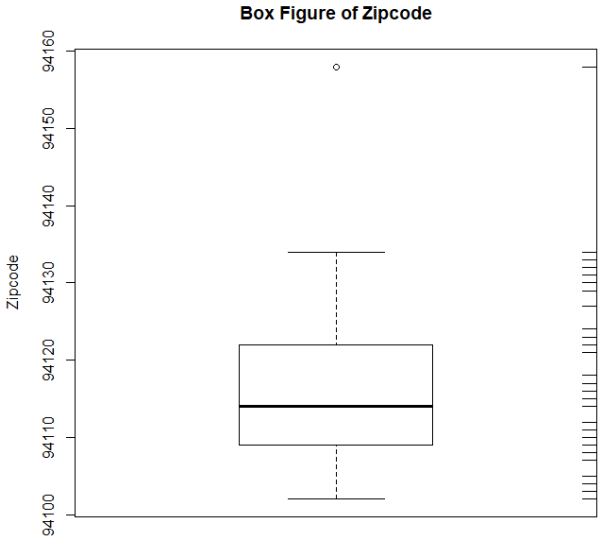
```r
#绘制QQ图
setwd(base_path)
dir.create("qqFigure")
qqPlot_path = paste(base_path, "qqFigure/", sep="")
setwd(qqPlot_path)

for (i in 1:ncol(data))
    if (class(data[, i]) != "factor")
    {
        qqPlot(data[[i]], main=paste('Norm QQ Plot of', names(data[i])), ylab=names(data[i]))
        savePlot(paste("qqPlot_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

```r
#绘制盒图
setwd(base_path)
dir.create("boxPlot")
boxPlot_path = paste(base_path, "boxPlot/", sep="")
setwd(boxPlot_path)

for (i in 1:ncol(data))
    if (class(data[, i]) != "factor")
    {
        boxplot(data[[i]], main=paste('Box Figure of', names(data[i])), ylab=names(data[i]))
        rug(data[[i]], side=4)
        abline(h=mean(data[[i]], na.rm=T), lty=2)
        savePlot(paste("boxPlot_", gsub(".", "-", names(data[i]), fixed=TRUE), sep=""), type=c("jpg"))
    }
```

以 Zipcode 属性为例，其直方图、qq 图和盒图分别如下所示。







4.2 数据缺失的处理（数据集 2）

4.2.1 将缺失部分剔除

直接用 na.omit()函数剔除掉缺失的数据。

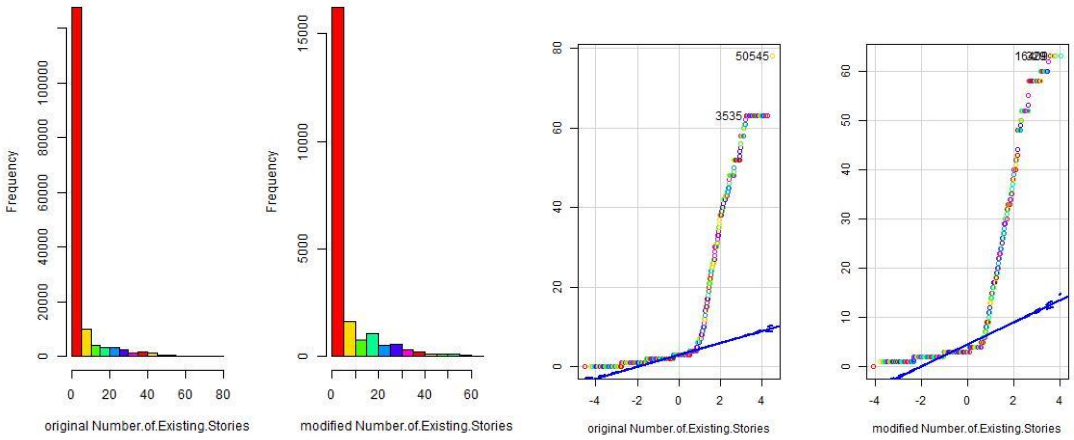
剔除缺失部分的代码如下：

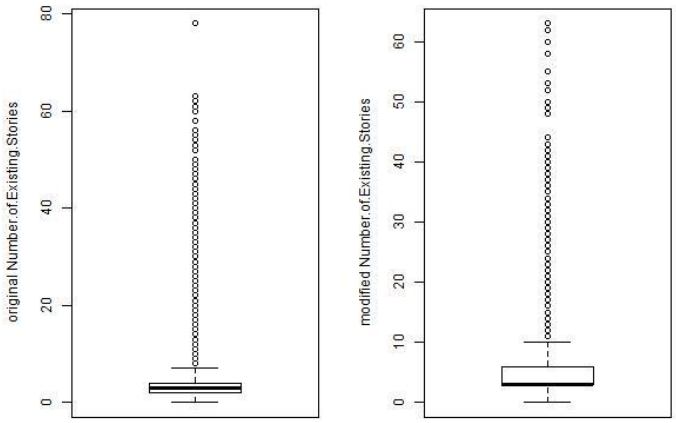
```
#将缺失部分剔除
data_omit1 = na.omit(data_omit1)
```

对比剔除缺失数据前后两者的维度：

```
> dim(data)
[1] 198900     43
> dim(data_omit1)
[1] 21683    43
```

对比剔除缺失数据前后两者的直方图、qq 图和盒图





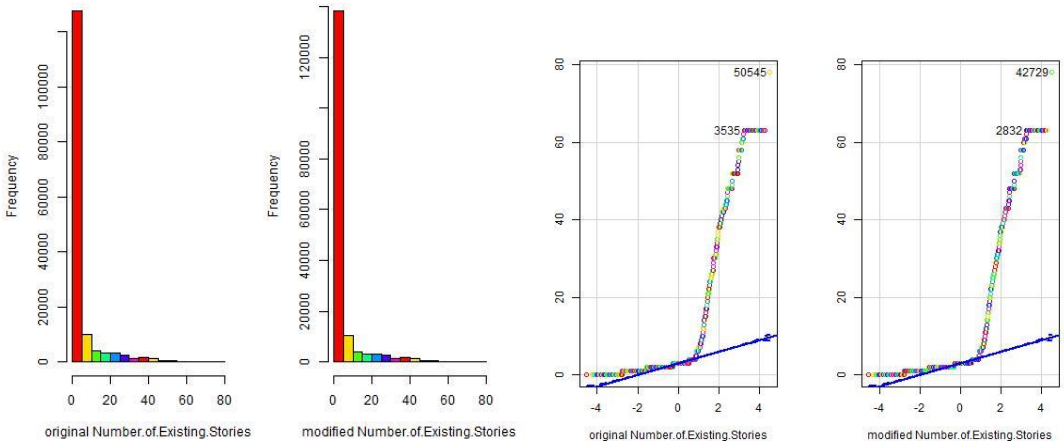
从直方图来看，剔除数据前后差异较大，特别是红色条柱的长度变化较大，qq 图和盒图的也差异较大。

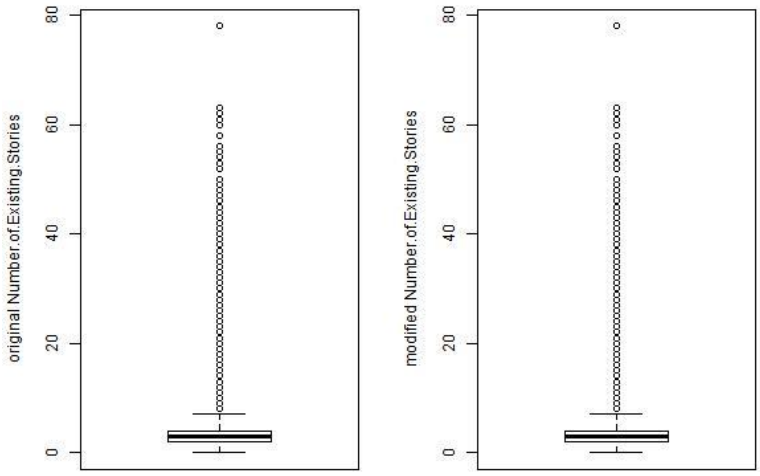
4.2.2 用最高频率来填补缺失值

对于数值属性的数据，用其中位数来填补缺失值；对于标签属性的数据，用其众数来填补缺失值。首先利用 manyNAs(data, nORp)函数来查找数据框 data 中缺失值过多（≥缺失比例 nORp）的行，nORp 默认为 0.2，即缺失值个数≥列数的 20%。然后利用 DMwR 包中的 centralImputation 函数来填补数据。核心代码如下：

```
#用最高频率来填补缺失值
data_omit2 = data[-manyNAs(data),]
data_omit2 = centralImputation(data_omit2)
```

以 Number.of.Existing.Stories 属性为例，分别对比填补缺失值前后的直方图、qq 图、盒图





对比可知，处理前后条形图变化较大，qq 图和盒图变化很小，故比前一种方法更准确。

4.2.3 通过属性的相关关系来填补缺失值

先探索变量之间的相关关系，找到相关性较高的两个变量后，在寻找他们之间的线性回归关系，最后通过线性回归关系计算缺失值进行填补。

先得到数据类型为 numeric 的属性的索引值，先通过 cor()函数来产生这些属性之间的相关值矩阵，设定参数 use=”complete.obs”可以使 R 在计算相关值时忽略含有 NA 值的样本，然后用 symnum()函数来改善结果的输出形式，得到结果如下：

```
> symnum(cor(data[, c(x)],use="complete.obs"))
                          P.T S.N U N..E N..P Es.C R.C E.U P.U Pl E.C. P.C S.D Z R.I
Permit.Type               1
Street.Number                 1
Unit                              1
Number.of.Existing.Stories          1
Number.of.Proposed.Stories          B    1
Estimated.Cost                            1
Revised.Cost                              B    1
Existing.Units                      .    .         1
Proposed.Units                      .    .         B   1
Plansets                                               1
Existing.Construction.Type          ,    ,         .   .      1
Proposed.Construction.Type          ,    ,         .   .      B    1
Supervisor.District                                                 1
Zipcode                             .    .                   .    .       1
Record.ID                                                                     1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

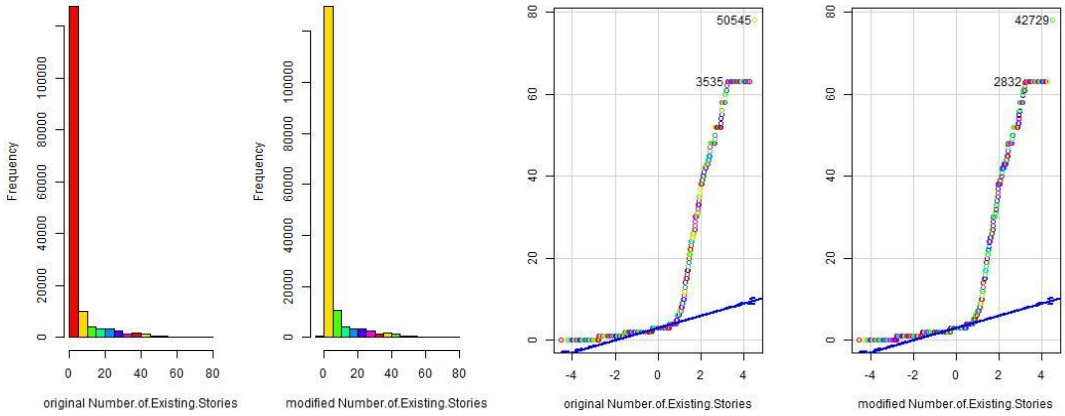
拿 Number.of.Existing.Stories 和 Number.of.Proposed.Stories 举例来说，它们之间的相关度超过 0.95。用 lm 函数来得到两者之间的相关关系如下。然后利用此回归关系填补 Number.of.Existing.Stories 属性的缺失值。

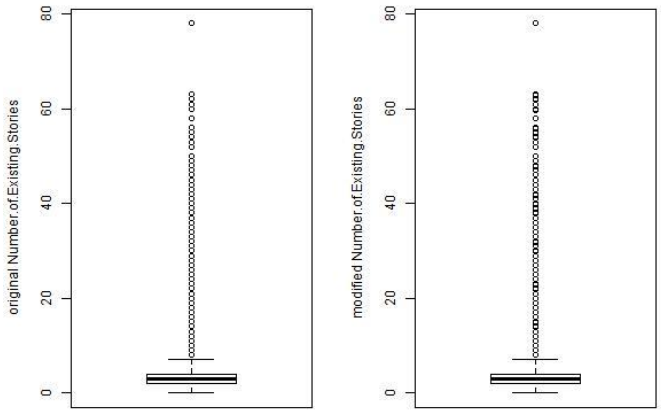
```
> lm(formula = Number.of.Existing.Stories~Number.of.Proposed.Stories, data = data)

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

Coefficients:
                (Intercept)  Number.of.Proposed.Stories
                    -0.0171                      0.9968
```

此部分的核心代码如下：

```
#通过属性的相关关系来填补缺失值
x<-vector(mode="numeric",length=0)
j = 1
for(i in 1:ncol(data))
    if(class(data[,i]) != "factor")
    {
        x[j] <- i
        j <- j+1
    }

symnum(cor(data[, c(x)],use="complete.obs"))
lm(formula = Number.of.Existing.Stories~Number.of.Proposed.Stories, data = data)
data_omit3 = data[-manyNAs(data),]
fillNE <- function(NP){
    if(is.na(NP))
        return(NA)
    else
        return (-0.0171+0.9968*NP)
}
data_omit3[is.na(data_omit3$Number.of.Existing.Stories), 'Number.of.Existing.Stories'] <-
        sapply(data_omit3[is.na(data_omit3$Number.of.Existing.Stories),'Number.of.Proposed.Stories'],fillNE)
```

对比两者的直方图、qq 图和盒图：





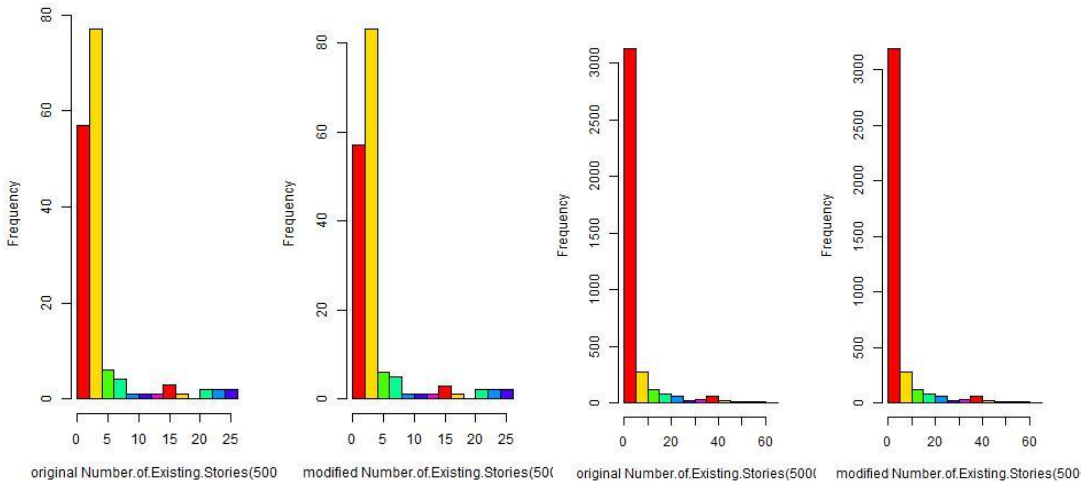
对比前后，可以发现直方图变化较前一种处理方法的变化大，qq 图和盒图变化较小，相对来说没有前一种方法准确合理一些。

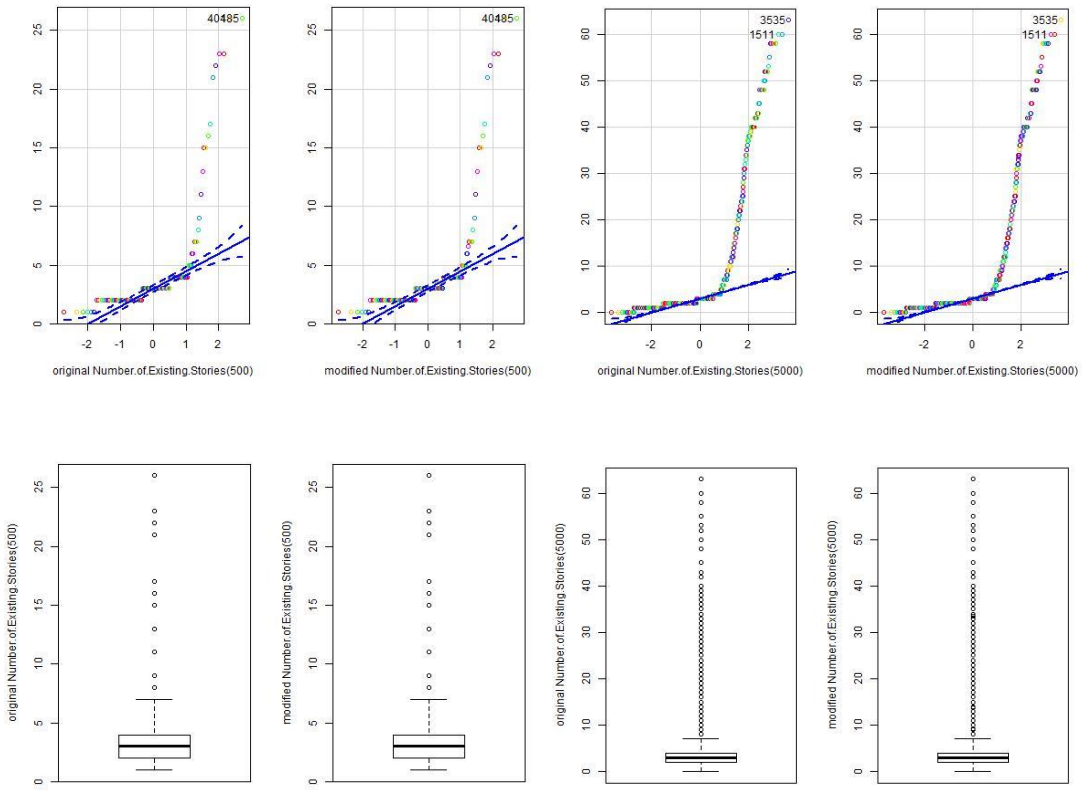
4.2.4 通过数据对象之间的相似性来填补缺失

可利用 KNN-均值来填充缺失值，其核心代码如下

```
#通过数据对象之间的相似性来填补缺失值
data_ori_500 = data[1:500, 1:25]
data_ori_5000 = data[1:5000, 1:25]
data_omit4_500 = knnImputation(data[1:500, 1:25], 10)
data_omit4_5000 = knnImputation(data[1:5000, 1:25], 10)
```

考虑到数据集太大，此处只对前 500 行和前 5000 行，前 25 列数据进行了处理。以 Number.of.Existing.Stories 为例，其直方图、qq 图和盒图对比如下，左侧对 500 行数据进行缺失处理的结果，右侧为对 5000 行数据进行缺失处理的结果。





从图中可以看出，此方法处理前后，其直方图、qq 图和盒图变化均很小，所以交上述三种处理方法都准确合理。