数据挖掘作业一报告

数据探索性分析与数据预处理

姓名：赵赫　　学号：**2120171103**

一、 问题描述

本次作业中，将对 2 个数据集进行探索性分析与处理。

分析和处理内容包括数据可视化和摘要、数据缺失的处理两部分。

- 在数据摘要任务中，对于数据集中的标称属性，给出每个可能取值的频数；对与数据集中的数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数。
- 在数据可视化任务中，对于数据集中的数值属性，分别（1）绘制直方图（2）绘制 q-q 图以检验其分布是否为正态分布（3）绘制盒图以对离群值进行识别。
- 在数据缺失处理任务中，观察数据集中的缺失数据，分析其缺失的原因，并分别使用四种策略对缺失值进行处理：（1）将缺失部分剔除（2）用最高频率值来填补缺失值（3）通过属性的相关关系来填补缺失值（4）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。

二、 数据说明

- **数据集 1：NFL Play-by-Play 2009-2017**
  该数据集共包含 101 个属性，407656 条数据记录。
- **数据集 2：San Francisco Building Permits**
  该数据集共包含 43 个属性，198900 条数据记录。

三、 数据分析过程

3.1 数据可视化和摘要

◆ **数据摘要：**

由于原始数据中的属性列个数较多，且属性类型（标称属性、数值属性）并不统一，因此为方便数据的处理和读取，采用 MySQL 数据库，分别

将两个表中的数据存入 MySQL 表中。对于 NFL play by play 数据集，其中的 desc 属性对应的值为文本描述，对此次数据分析建模不产生作用和影响，因此没有导入这一属性到数据库中，同理，对于 Building Permits 数据集中的 description 属性也没有导入到数据库。

经过对**数据集 1** 的 101 个属性进行人工识别，其中包含标称属性 56 个，分别为：

Date、GameID、Time、SideofField、GoalToGo、FirstDown、Posteam、DefensiveTeam、PlayAttempted、Sp、Touchdown、ExPointResult、TwoPointConv、DefTwoPoint、Safety、Onsidekick、PuntResult、PlayType、Passer、Passer_ID、PassAttempt、PassOutcome、PassLength、QBHit、PassLocation、InterceptionThrown、Interceptor、Rusher、Rusher_ID、RushAttempt、RunLocation、RunGap、Receiver、Receiver_ID、Reception、ReturnResult、Returner、BlockingPlayer、Tackler1、Tackler2、FieldGoalResult、Fumble、RecFumbTeam、RecFumbPlayer、Sack、Challenge_Replay、ChalReplayResult、Accepted_Penalty、PenalizedTeam、PenaltyType、PenalizedPlayer、HomeTeam、AwayTeam、Timeout_Indicator、Timeout_Team、Season

包含数值属性 45 个，分别为：

Drive、Qtr、Down、TimeUnder、TimeSecs、PlayTimeDiff、Yrdln、Yrdline100、Ydstogo、Ydsnet、Yards_Gained、AirYards、YardsAfterCatch、FieldGoalDistance、Penalty_Yards、PosTeamScore、DefTeamScore、ScoreDiff、AbsScoreDiff、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

经过对**数据集 2** 的 43 个属性进行人工识别，其中包含标称属性 33 个，分别为：

Permit_Number、Permit_Type、Permit_Type_Definition、Permit_Creation_Date、Block、Lot、Street Number、Street Number Suffix、Street Name、Street Suffix、Unit、Unit Suffix、Current Status、Current Status Date、Filed Date、Issued Date、Completed Date、First Construction Document Date、Structural Notification、Voluntary Soft_Story Retrofit、Fire Only Permit、Permit Expiration Date、Existing Use、Proposed Use、TIDF Compliance、Existing Construction Type、Existing Construction Type Description、Proposed Construction Type、Proposed Construction Type Description、Site_Permit、

Neighborhoods_Analysis_Boundaries、Zipcode、Record_ID

包含数值属性 8 个，分别为：

Number of Existing Stories、Number of Proposed Stories、Estimated Cost、Revised Cost、Existing Units、Proposed Units、Plansets、Supervisor District

两个数据集的标称属性、数值属性名称列表文件保存在目录：“./data/”:

-Homework1

-data

-dataset1

-标称属性列名.txt

-数值属性列名.txt

-dataset2

-标称属性列名.txt

-数值属性列名.txt

首先调用 Data 类中的 process_nom_features()函数对两个数据集的标称属性进行处理，返回该属性对应数据的各个取值及其频数。调用方法如下：

```
data = Data()
# 分别处理两个数据集的标称属性
data.process_nom_features(data.dataset1_nom_feature_list, dataset1_table_name)
data.process_nom_features(data.dataset2_nom_feature_list, dataset2_table_name)
```

获得的对应结果位于“./results/dataset*/标称属性”目录下，文件格式为：

```
ExPointResult.txt:                  # 结果文件名
    Feature Name: ExPointResult     # 属性名称
    Value Num: 5                    # 不同的取值个数
    NA,397546                       # 缺失值，个数
    Blocked,81                      # 取值 1，个数
    Missed,172                      # 取值 2，个数
    Aborted,7                       # ……
    Made,9850
```

然后调用 Data 类中的 process_num_features()函数对两个数据集的数值属性进行处理，在该函数中，首先创建 NumProcessor 类的对象 processor 用于处理数值属性，并调用 processor 的 pre_process()函数获取描述数据分布的缺失值个数、最大值、最小值、均值、中位数以及两个上下四分位数。调用方法如下：

```
# 分别处理两个数据集的数值属性
data.process_num_features(data.dataset1_num_feature_list, dataset1_table_name)
data.process_num_features(data.dataset2_num_feature_list, dataset2_table_name)
```

```
processor = NumProcessor(feature_name, values)
# 获取描述数据集的几个数值
missing_num, max_num, min_num, average_num, median_num, quartile1, quartile2 = processor.pre_process()
```

获得的对应结果位于“./results/dataset*/数值属性”目录下，文件格式为：

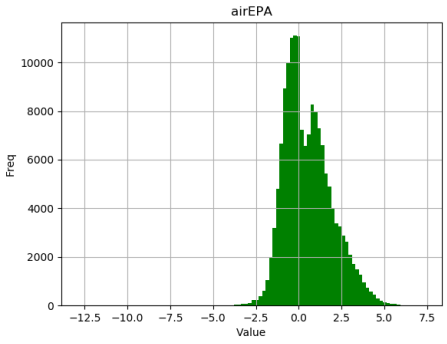
AwayTimeouts_Remaining_Pre.txt: # 结果文件名
Feature Name: AwayTimeouts_Remaining_Pre # 属性名称
Missing Num: 0 # 缺失值个数
Max Num: 3.0 # 最大值
Min Num: -1.0 # 最小值
Average Num: 2.5172596502933846 # 平均值
Median Num: 3.0 # 中位数
Quartile Num: 2.0, 3.0 # 下四分位数，上四分位数

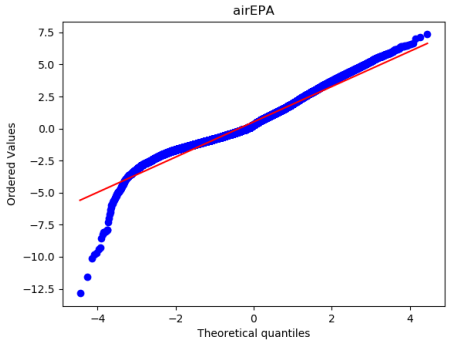
◆ **数据可视化：**

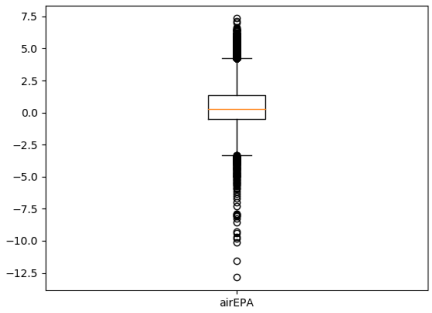
对于两个数据集的全部数值属性，在 process_num_features()函数中分别调用 processor 对象的 draw_histogram()、draw_qq_plot()、draw_box()函数绘制直方图、q-q 图、盒图。调用方法如下：

```
# 为数据画直方图
processor.draw_histogram(processor.new_value_set, feature_name, fig_path)
# 为数据画q-q图
processor.draw_qq_plot(processor.new_value_set, feature_name, fig_path)
# 为数据画盒图
processor.draw_box(processor.new_value_set, feature_name, fig_path)
```

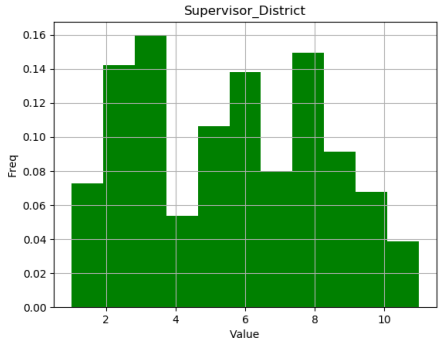
绘图结果图片分别位于
“./results/dataset*/figure/histogram”、
“./results/dataset*/figure/q-q”、
“./results/dataset*/figure/box”
目录下，图片以（属性名.png）命名。
以数据集 1 的 airEPA 属性举例，三类图结果为：

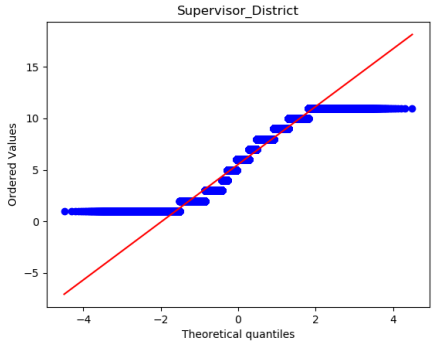


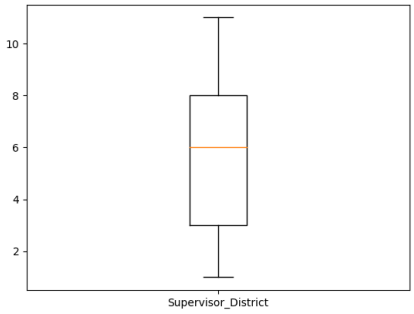




以数据集 2 的 Supervisor_District 属性举例，三类图结果为：







3.2 数据缺失处理

为尝试用四种不同方法对缺失值进行填补，对于这一任务选取了数据集 1 的 5 个属性列（score_dif, no_score_prob, safety_prob, WPA, win_prob）为例，数据缺失原因可能为记录数据时漏填、因为某些属性关联依赖导致的该属性无对应值等。分别采用（1）剔除缺失值（2）用最高频率值填补（3）利用属性相关关系进行填补（4）利用数据对象相似度 四种方法填补缺失数据。

◆ **缺失值填补：**

借助 python3.5.2 下 fancyimpute 包进行数据缺失值填补，fancyimpute 不仅高效实现了利用一些基本数值（0 值、均值、众数、中位数）进行数据填补的 simple_fill 方法，还封装了基于属性相关关系进行数据填补的的 R 语言 MICE 工具包，以及基于 KNN 进行数据对象相似度聚类计算进行数据填补的函数，可以方便快速地辅助数据缺失值填补。Fancyimpute 包地址链接为：

https://pypi.python.org/pypi/fancyimpute。

调用 Data 类的 impute_missing_values()方法开始对缺失值处理，在该方法中进一步调用 NumProcessor 类 processor 对象的 impute_missing_values(value_set，strategy)方法进行缺失值填充，该方法接收两个参数：value_set 为待填充的数据矩阵，strategy 取值在[1,2,3,4]中，代表用第几种方法进行缺失值填补。函数调用方法如下：

```
# 选取dataset1中的五个属性进行填充
data.impute_missing_values()
```

```
processor = NumProcessor()
# 分别用四种策略对缺失值进行填充
for strategy in [1,2,3,4]:
    strategy_out_path = os.path.join(out_path, "strategy"+str(strategy))
    feature_values = processor.impute_missing_values(values, strategy)
```

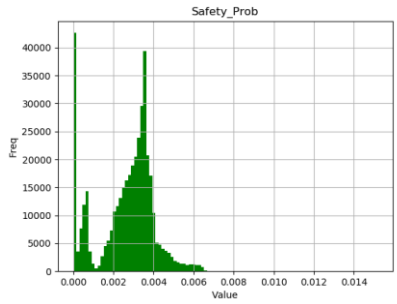
◆ **可视化对比：**

为经过填补的数据分别绘制直方图、q-q 图、盒图，并将绘图结果与原始数据进行对比。同样调用 processor 对象的 draw_hitogram()、draw_qq_plot()、draw_box()函数进行绘图，函数调用方法如下：

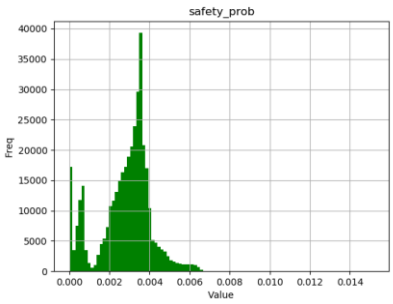
```
# 分别对该feature画三种图
processor.draw_histogram(value_set, feature_name, strategy_out_path)
processor.draw_qq_plot(value_set, feature_name, strategy_out_path)
processor.draw_box(value_set, feature_name, strategy_out_path)
```

绘图结果图片文件保存于“./results/imputed_figures/strategy*(*=1/2/3/4)”目录下。

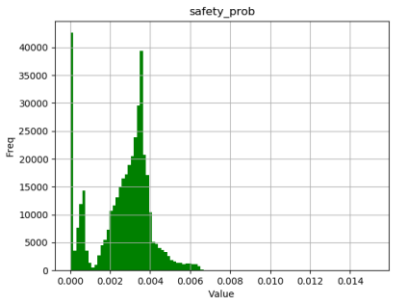
以 Safety_prob 属性为例展示可视化对比结果：



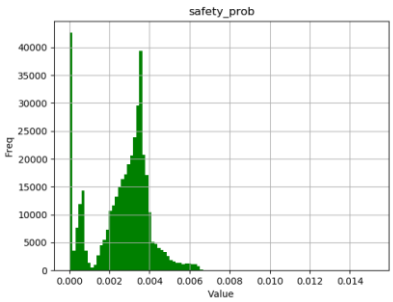
原始直方图



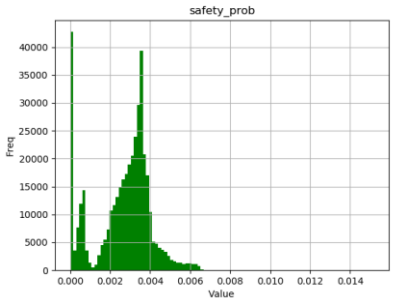
剔除缺失值直方图



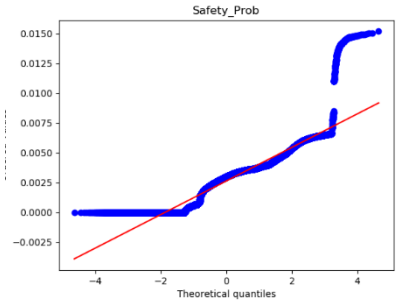
最高频值填补直方图



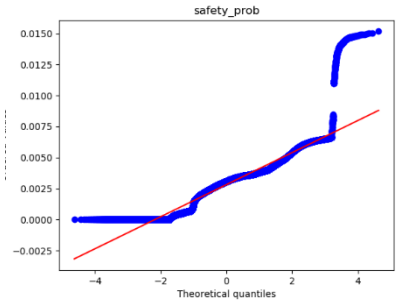
属性相关关系填补直方图



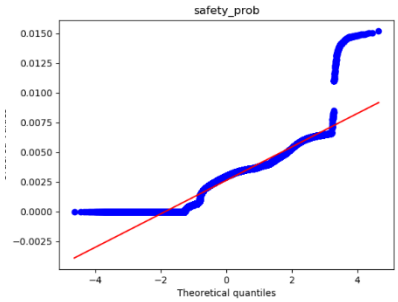
数据对象相似性填补直方图



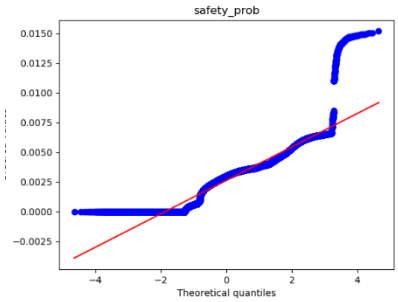
原始 q-q 图



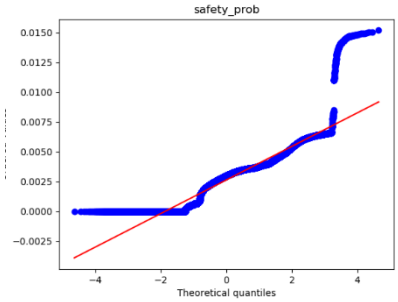
剔除缺失值 q-q 图



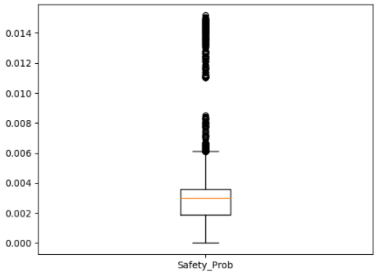
最高频值填补 q-q 图



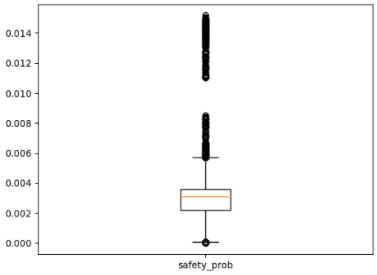
属性相关关系填补 q-q 图



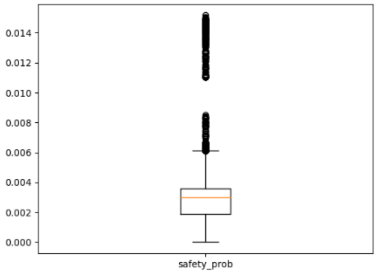
数据对象相似性填补 q-q 图



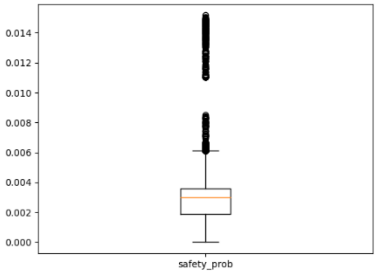
原始盒图



剔除缺失值盒图



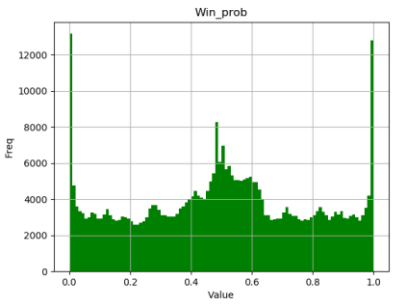
最高频值填补盒图



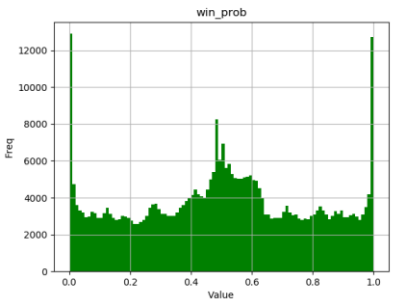
属性相关关系填补盒图

数据对象相似性填补盒图

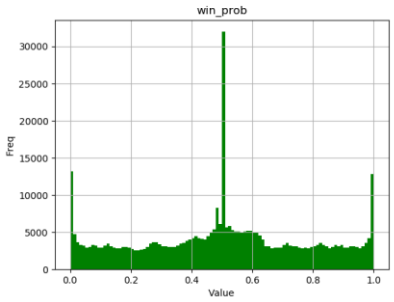
以 Win_prob 属性为例展示可视化对比结果：



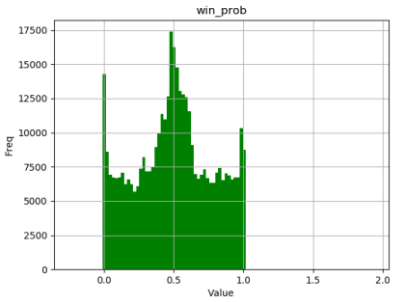
原始直方图



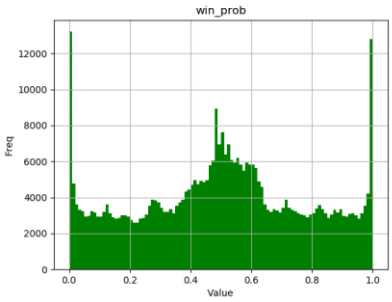
剔除缺失值直方图



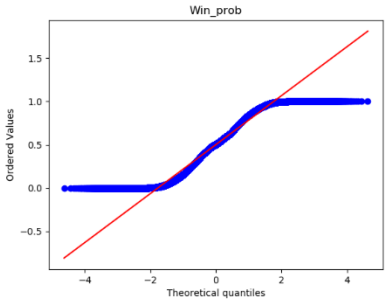
最高频值填补直方图



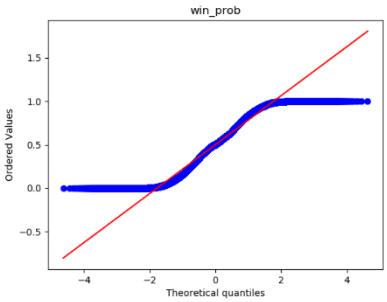
属性相关关系填补直方图



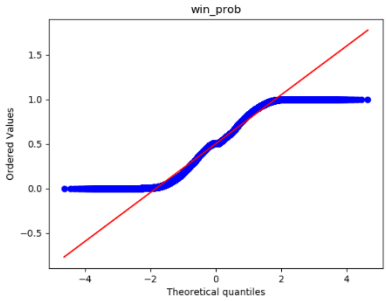
数据对象相似性填补直方图



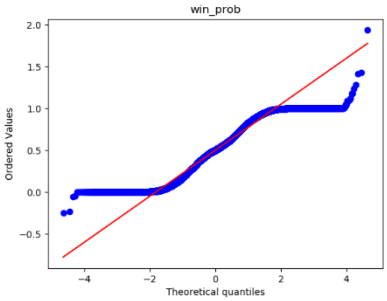
原始 q-q 图



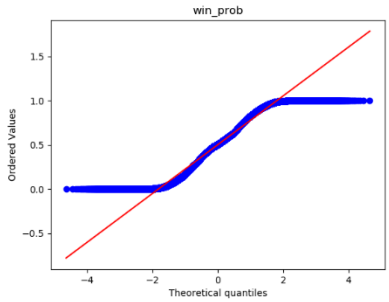
剔除缺失值 q-q 图



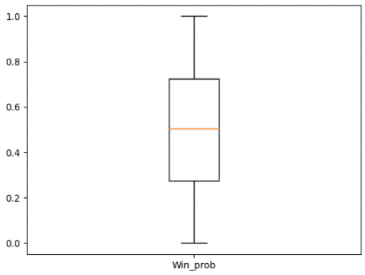
最高频值填补 q-q 图



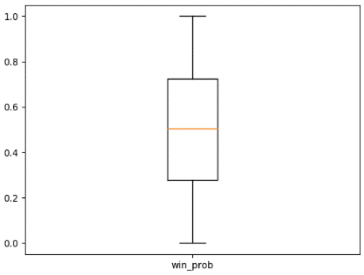
属性相关关系填补 q-q 图



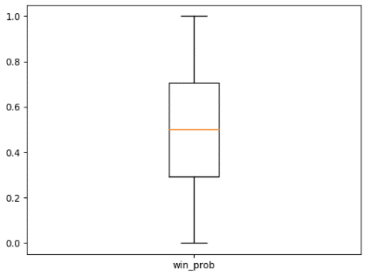
数据对象相似性填补 q-q 图



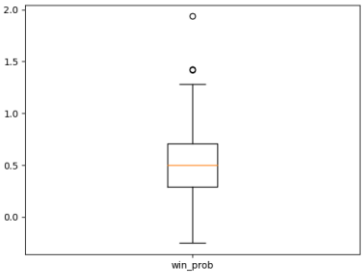
原始盒图



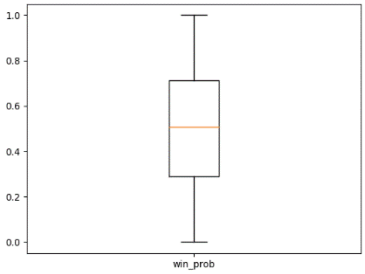
剔除缺失值盒图



最高频值填补盒图



属性相关关系填补盒图



数据对象相似性填补盒图