**缺省值**

**缺省值的检测**

pandas中的None和NaN表示缺失值

|  |  |
| --- | --- |
| 检测缺省值的方法 | 说明(四个方法都是返回与原对象形状相同的布尔矩阵) |
| isnull() | 若元素位置返回True,则该元素是缺省值 |
| notnull() | 若元素位置返回False,则该元素是缺省值 |
| isna() | 若元素位置返回True,则该元素是缺省值 |
| notna() | 若元素位置返回False,则该元素是缺省值 |

**其他函数**

df.apply(lambda x: sum(x.isnull())/len(x),axis= 0) #缺失比例

**缺省值的处理**

**删除缺省值**

DataFrame.dropna(axis=0,how='any',thresh=None,subset=None,inplace=False)

axis：0或index表示删除包含缺省值的行；1或columns表示删除包含缺省值的列

how：any表示删除任何有NaN的行或列；all表示删除全部都是NaN的行或列

thresh：表示保留至少有N个非NaN值的行或列

subset：删除指定列的缺省值

inplace：True表示会修改原数据；False表示会修改副本数据

**填充缺省值**

DataFrame.fillna(value=None,method=None,axis=None,inplace=False,limit=None,downcast=None)

value：表示填充的值

method：pad或ffill表示将最后一个有效值向后传播；backfill或bfill表示将最后一个有效值向前传播

axis：0或index表示删除包含缺省值的行；1或columns表示删除包含缺省值的列

limit：表示连续填充的最大数目

#Mileage是df对象的一个字段

df.Mileage.fillna(df.Mileage.mean()) # 年龄用均值填补

df.Mileage.fillna(df.Mileage.median()) #中位数填补

df.Exterior\_Color.fillna(df.Exterior\_Color.mode()[0]) #众数填补

**插补缺省值**

DataFrame.interpolate(method='linear',axis=0,limit=None,inplace=False,limit\_direction=None,limit\_area=None,downcast=None)

method：linear表示用线性插值填充；time表示根据时间长短进行填充(用于索引为时间的)；index和values表示用索引的实际值进行填充；nearest表示用最邻近插值法进行填充；barycentric表示用重心坐标插值法进行填充

limit：表示连续填充的最大数目

limit\_direction：forward表示向前填充；backforward表示向后填充；both：表示向前又向后

**重复值**

**重复值的检测**

DataFrame.duplicated(subset=None,keep='first')

subset：表示识别重复项的列索引或列索引序列

keep：first表示删除重复项,仅保留第一次出现的数据项；last表示删除重复项,仅保留最后一次出现的数据项；False表示所有相同的数据都标记为重复项

**重复值的处理**

DataFrame.drop\_duplicates(subset=None,keep='first',inplace=False,ignore\_index=False)

subset：表示识别重复项的列索引或列索引序列

keep：first表示删除重复项,仅保留第一次出现的数据项；last表示删除重复项,仅保留最后一次出现的数据项；False表示所有相同的数据都标记为重复项

inplace:True表示会修改原数据；False表示会修改副本数据

ignore\_index:表示是否对返回值进行重新排序

**异常值**

**异常值的检测**

**标准差法**

xbar = df.Price.mean()

xstd = df.Price.std()

print('标准差法异常值上限检测：\n',any(df.Price> xbar + 2.5 \* xstd))

print('标准差法异常值下限检测：\n',any(df.Price< xbar - 2.5 \* xstd))

**用箱形图检测异常值**

DataFrame.boxplot(column=None,fontsize=None,rot=0,grid=True,return\_type=None)

column：表示检测的列名

fontsize：表示坐标轴文字的大小

rot：表示坐标轴旋转的角度

grid：表示箱型图的窗口大小

return\_type：axes表示符合matplotib的axes绘画对象；dict表示返回一个字典，其值是matplotib的Line对象；both表示返回(axes,dict)的元组

Q1 = df.Price.quantile(q = 0.25)

Q3 = df.Price.quantile(q = 0.75)

IQR = Q3 - Q1

print('箱线图法异常值上限检测：\n',any(df.Price > Q3 + 1.5 \* IQR))

print('箱线图法异常值下限检测：\n',any(df.Price < Q1 - 1.5 \* IQR))

# 箱线图法异常值上限检测：

# True

# 箱线图法异常值下限检测：

# False

df.boxplot(column='Price')#显示箱型图

**正态分布检测**

stats.kstest(column，moth,data)

column：表示检测的列

moth：表示检测方法(norm表示正态分布)

data：表示(平均值,标准差)的元组

import scipy.stats as stats

u=df['value'].mean()

std=df['value'].std()

stats.kstest(df['value'],'norm',(u,std))#检测是否符合正态分布

**异常值的处理**

**删除异常值**

DataFrame.drop(labels=None,axis=0,index=None,columns=None)

labels：表示要删除的列索引或行索引(可删除多个)

axis：0或index表示删除包含缺省值的行；1或columns表示删除包含缺省值的列

index：表示要删除的行

columns：表示要删除的列

**替换异常值**

DataFrame.replace(to\_place=None,value=None,inplace=False,method='pad)

to\_place：表示被替换的值

value：表示用来替换的值

inplace:True表示会修改原数据；False表示会修改副本数据

method：pad或ffill表示将最后一个有效值向后传播；backfill或bfill表示将最后一个有效值向前传播