|  |
| --- |
| # -\*- coding: utf-8 -\*- |
|  | """ |
|  | Spyder Editor |
|  |  |
|  | This is a temporary script file. |
|  | """ |
|  | # -\*- coding: utf-8 -\*- |
|  |  |
|  | #第4章实训1 |
|  | import pandas as pd |
|  | import numpy as np |
|  |  |
|  | Mas=pd.read\_csv("C:\Users\3SXL\_21128\Desktop\ch4\ch4-实训数据\Training\_Master.csv",encoding="gbk") |
|  | User=pd.read\_csv("C:\Users\3SXL\_21128\Desktop\ch4\ch4-实训数据\Training\_Userupdate.csv",encoding="gbk") |
|  | Log=pd.read\_csv("C:\Users\3SXL\_21128\Desktop\ch4\ch4-实训数据\Training\_LogInfo.csv",encoding="gbk") |
|  | print("主表的维度为：",Mas.ndim) |
|  | print("主表的形状为：",Mas.shape) |
|  | print("主表的占用内存信息为：",Mas.memory\_usage()) |
|  |  |
|  | print("用户信息表的维度为：",User.ndim) |
|  | print("用户信息表的形状为：",User.shape) |
|  | print("用户信息表的占用内存信息为：",User.memory\_usage()) |
|  |  |
|  | print("登录信息表的维度为：",Log.ndim) |
|  | print("登录信息表的形状为：",Log.shape) |
|  | print("登录信息表的占用内存信息为：",Log.memory\_usage()) |
|  |  |
|  | print(Mas.describe())#描述性统计 |
|  | #定义一个函数剔除全为空值的列和标准差为0的列 |
|  | def dropNullStd(data): |
|  | beforelen = data.shape[1] |
|  | colisNull = data.describe().loc['count'] == 0 |
|  | for i in range(len(colisNull)): |
|  | if colisNull[i]: |
|  | data.drop(colisNull.index[i],axis=1,inplace=True) |
|  |  |
|  | stdisZero = data.describe().loc['std'] == 0 |
|  | for i in range(len(stdisZero)): |
|  | if stdisZero[i]: |
|  | data.drop(stdisZero.index[i],axis=1,inplace=True) |
|  | afterlen = data.shape[1] |
|  | print('剔除的列的数目为：',beforelen-afterlen) |
|  | print('剔出后数据的形状为：',data.shape) |
|  | dropNullStd(Mas)#使用dropNullStd函数对信息表操作 |
|  | dropNullStd(User) |
|  | dropNullStd(Log) |

|  |
| --- |
| #第4章实训2 |
|  |  |
|  | pon = pd.read\_csv('data\ch4\实训数据\Training\_LogInfo.csv',sep=',',encoding='gbk') |
|  | pup = pd.read\_csv('data\ch4\实训数据\Training\_Userupdate.csv',sep=',',encoding='gbk') |
|  |  |
|  | pon['Listinginfo1'] = pd.to\_datetime(pon['Listinginfo1']) |
|  | pon['LogInfo3']= pd.to\_datetime(pon['LogInfo3']) |
|  | pup['ListingInfo1']= pd.to\_datetime(pup['ListingInfo1']) |
|  | pup['UserupdateInfo2']= pd.to\_datetime(pup['UserupdateInfo2']) |
|  |  |
|  | print(pon['Listinginfo1'].dtypes) |
|  | print(pon['LogInfo3'].dtypes) |
|  | print(pup['ListingInfo1'].dtypes) |
|  | print(pup['UserupdateInfo2'].dtypes) |
|  |  |
|  | def ymw(data): |
|  | year = [i.year for i in data] |
|  | month = [i.month for i in data] |
|  | week = [i.week for i in data] |
|  | print(year[:5]) #提取信息表前5条数据的年份信息 |
|  | print(month[:5])#提取信息表前5条数据的月份信息 |
|  | print(week[:5])#提取信息表前5条数据的星期信息 |
|  |  |
|  | ymw(pon['Listinginfo1']) |
|  | ymw(pon['LogInfo3']) |
|  | ymw(pup['ListingInfo1']) |
|  | ymw(pup['UserupdateInfo2']) |
|  |  |
|  |  |
|  | def userTimedelta(bdata,rdata): |
|  | timedelta = pd.to\_datetime(bdata) - pd.to\_datetime(rdata) |
|  | print('减去后的数据:',timedelta[:5]) |
|  | print('数据类型:',timedelta[:5]) |
|  |  |
|  | userTimedelta(pon['Listinginfo1'],pon['LogInfo3']) |
|  | userTimedelta(pup['ListingInfo1'],pup['UserupdateInfo2']) |
|  |  |
|  | #第4章实训3 |
|  | ponidgp = pon[['Idx','Listinginfo1','LogInfo1','LogInfo2','LogInfo3']].groupby(by='Idx') |
|  | pupidgp = pup[['Idx','ListingInfo1','UserupdateInfo1','UserupdateInfo2']].groupby(by='Idx') |
|  | print("最早更新时间，最晚更新时间为:",ponidgp[['LogInfo3']].agg(['max','min']).head()) |
|  | print("最早登陆时间，最晚登陆时间为:",pupidgp[['UserupdateInfo2']].agg(['max','min']).head()) |
|  | print("信息更新次数:",ponidgp.size()) |
|  | print("登陆次数:",pupidgp.size()) |

#实训四

|  |
| --- |
|  |
|  | import numpy as np |
|  | import pandas as pd |
|  | Master=pd.read\_csv("C:\\Users\\3SXL\_21152\\Desktop\\data\\Training\_Master.csv",encoding="gbk") |
|  | Userupdate=pd.read\_csv("C:\\Users\\3SXL\_21152\\Desktop\\data\\Training\_Userupdate.csv",encoding="gbk") |
|  | LogInfo=pd.read\_csv("C:\\Users\\3SXL\_21152\\Desktop\\data\\Training\_LogInfo.csv",encoding="gbk") |
|  | print("Master表长宽转换前的形状：",Master.shape) |
|  | print("Userupdate表长宽转换前的形状：",Userupdate.shape) |
|  | print("LogInfo表长宽转换前的形状：",LogInfo.shape) |
|  |  |
|  | CMaster=Master.T |
|  | print("Master表长宽转换后的形状：",CMaster.shape) |
|  | PMaster=pd.pivot\_table(Master,index="Idx") |
|  | print("使用pivot\_table后Master表的形状为：",PMaster.shape) |
|  | print("使用pivot\_table后Master表为：",PMaster.head()) |
|  |  |
|  | CRMaster=pd.crosstab(index=Master["Idx"],columns=Master["ListingInfo"],values=Master["ThirdParty\_Info\_Period2\_7"],aggfunc=np.sum) |
|  | print("使用crosstab后Master表为：",CRMaster.head()) |