Cheatography

Pandas Cheat Sheet

by cgeeeeh via cheatography.com/191000/cs/39717/

初始化

pd.Series(data={key:value,key:value},index,name)。index如果存在key里没有的值，value是NaN; Name是series的名字，在dataframe里一个series的名字是列名

pd.DataFrame(data,index,columns)。给数据用list是按行，比如[[row1][row2][row3]]，用dict是按列，如{key:[],key:[]}

CSV导入：pd.read_csv("data.csv", sheetname, skiprows, header, nrows, index_col, names,encoding)。sheetname似乎对csv文件不适用；skiprows按照【下标】跳过某些行，如range(0,6)，给的是下标，比如3就是跳过4行；header是第一行数据所在的【下标】，不建议使用因为中文有些字符包含\n；nrows是读取多少行；index_col是索引列的【下标】；names可以重命名列

属性

df.columns	df.index
df.axes	df.values
df.ndim	df.shape
df.size	df.empty
df.dtypes	df.str

df.T

df.index.get_level_values(0/"index_name")-读取多重索引中的一个

仅限dataframe：df.items()->(index,series)/-df.iterrows()->(index,series)

初探

df.describe()返回summary statistics/df.info()返回index&data type

df.head(5)/df.tail(5)

初探 (cont)

df.set_index(col,inplace=True)。将列设置成索引。col是列名，也可以是list（多个列名）

dfc=df.astype(dtype)改变数据类型

df.isnull()

df.corr()

Pandas显示设置

pd.set_option(,)。'max_colwidth' value的显示长度，默认50；'display.max_columns'+None显示所有列；'display.max_rows'+None显示所有行

链式赋值的警告：pd.options.mode.chained_assignment = None

预处理

转数据格式：df["Colname"].astype()

字符串转时间再转回来：df["Date"]=df["Date"].apply(lambda x: pd.Timestamp(str(x)).strftime("%Y-%m-%d"))

转list：series.to_list()

判断空值：=df.isnull().sum()/=df.isna().sum()按列求和，返回一个series/=df.notnull()

填空值：df["colname"].fillna(df['colname'].mode()/“ffill”,inplace=True)；ffill是填写上一个非空值

删除空值：dv1.dropna(inplace=True)

列重命名：df.rename(columns={0:"sales"},-inplace=True)

排序：df.sort_values(by=["Colname1","Colname2"],ascending=True,inplace)

预处理 (cont)

排序：df["Name"].rank(axis, method, numeric_only, na_option, ascending, pct)。axis=0对行排序，axis=1对列排序; method-有min,max,average,first,dense; numeric_only决定是否只对数值排序，false则只对非数值排序; na_option有keep, top, bottom

重新组合：df.reshape(x,x)

堆叠：stack()。把columns堆在index里面，形成多层索引

透视表：pivot("col1","col2")。把sheet里的数据转换成透视表，第一个是索引，第二个-是列。随后列索引会有两层，因此需要只取出第二层

拼接：pd.concat([df1,df2], axis, ignore_index), axis默认0，0是沿index（上下），1是沿着column（左右）。沿着index concat，df2多的列新建，df1有的拼接在下面

join：pd.merge(dv1,df_rates,on,how)，on没有指定的时候，就是两个df的列的交集，-how默认是inner

删除某几列的重复值：df.drop_duplicates(subset=['colname'])

将dummy们转化：pd.get_dummies(df, columns=["col1","col2"])

计算

max，min索引：df["Test1"].idxmax()/idxmin()

count(), mean(), var(), std(), median(), mode(), sum(), unique(), cov()

df.groupby(['A'])['B','C'].agg(np.mean)/.count()

Cheatography

Pandas Cheat Sheet

by cgeeeeh via cheatography.com/191000/cs/39717/

添加数据

行：loc

list_row = ["Hyperion", 27000, "60days", 2000] df.loc[len(df)] = list_row

行：df.append()

new_row = {'Courses':'Hyperion', 'Fee':-24000, 'Duration':'55days', 'Discount':1800} df2 = df.append(new_row, ignore_index-=True)

行：append with named index

df2 = df.append(pd.DataFrame([new_row],-index=['7'],columns=df.columns))

行：append，series。但是竖着的series append进dataframe后变成行了

df2 = df.append(pd.Series(new_row, index=-df.columns, name='7'))

行：pd.concat()

new_row = pd.DataFrame({'Courses':'Hyperion', 'Fee':24000, 'Duration':'55days', 'Discount':1800}, index=[0]) df2 = pd.concat([new_row,df.loc[:]]).reset_index(drop=True)

行：df.loc[]（可以把index一起加进去么？）

df.loc['7', :] = ['Hive',25000,'45days',1800]

行：pd.concat

df_row_reindex = pd.concat([df1, df2], ignore_index)。ignore_index=True则重新索引，axis=0上下堆，axis=1左右堆

pieces = {'x': df1, 'y': df2}；df_piece = pd.concat(pieces)

列：df1["colname"]=[30,52,50,28]

用索引查缺补漏：ser2.combine_first(ser-1)。ser1里的数据，如果索引ser2没有则补-充，如果有则不变

删数据

del df1["Colname"]

df.drop(index如["colname","colname2"]/[3,-4,6],axis,inplace)。index可以是行或列索引，axis默认为0删除行，1删除列

X=df.drop(columns="Creditability")

空值：df.dropna(axis=0, inplace=True)。a-xis默认为0删行，1删列

重复：df.drop_duplicates(subset=None,ke-ep,inplace,ignore_index)。subset接受索引，是个list；keep有first，last，False；i-gnore_index=False，True则会重排序

查数据

单列：df["col_name"]->series/df[["col_n-ame"]]->dataframe，df.loc[colname]

多列：df.loc[:,["Test2","Test3"]]/df.loc[:,"-Test2":"Test3"]/df.loc[: , [col for col in df.columns if 'Random' in col]]

单行：df.loc['John']/df.loc[2]；df.iloc[1]。-loc和iloc的区别是，查阅索引和下标

多行：df.loc[[0,1]]/df.loc[['Peter','Mary']]/df.l-oc[[2:4]]/df.loc['Peter':'Mary']

某个/片单元格：df.loc[["peter","Marry"],["-test1","test2"]]/df.iloc[1:2,1:3]

dv1.loc[dv1["MMM"]==mths[i],"MM"]="-aa"单个值也可以

依据列查询行：df.loc[~ (df["Name"]==xxx) | (df.Name.isin(["John","Peter"])), ["A","-B",""C]]

依据行查询列：df.loc[:,df.isin([54,56]).any-()]/df.loc[: , [(df[col] == 30).any() for col in df.columns]]

依据datatype查询：df.select_dtypes('objec-t'/'number')

查数据 (cont)

df[][]被允许，但是第一个参数是行的索引切片（如1：2），第二个参数是列（list）。-df[1]["x"]的表述是不允许的

连接

只有一列同名列：pd.merge(frame1,frame2)

多列同名：pd.merge(df1,df2,on="colname")

连接不同名的两列：pd.merge(df1,df2,left-_on="col1",right_on="col2")

连接多列：pd.merge(df1, df2, on=["col1","-col2"], how)。how有right,left,outer

INNER JOIN：only the set of records that match in both A and B

LEFT JOIN：a complete set of records from A (left DataFrame), with the matching records (where available) in B (right DataFrame). If there is no match, the left side will contain null.

OUTER JOIN：combines the results of both the left and the right outer joins

索引连接：pd.merge(df1,df2,right_index-=True,left_index=True)

join with index：df1.join(df2, lsuffix, rsuffix,)

join with key col：df.set_index('key').join(o-ther.set_index('key'))/df.join(other.set_inde-x('key'), on='key')

输出到文件

df.to_csv('scores.csv',encoding="utf-8")

from pandas import ExcelWriter writer = ExcelWriter('score.xlsx') df.to_excel(wri-ter,'Sheet10') writer.save()

dictionary = df.to_dict()

string = df.to_string()

By **cgeeeeh**
cheatography.com/cgeeeeh/

Not published yet.
Last updated 4th December, 2023.