**日期格式数据处理**

1.

Pandas中使用to\_datetime()方法将文本格式转换为日期格式

2.

dataframe数据类型如果为datetime64,可以使用dt方法取出年月日等

3.

对于时间差数据,可以使用timedelta函数将其转换为指定时间单位的数值

4.

时间差数据,可以使用dt方法访问其常用属性

import numpy as np

import pandas as pd

import os

os.chdir(r'C:\Users\chell\Desktop')

df=pd.read\_csv('baby\_trade\_history.csv',encoding='gbk',dtype={'user\_id':str})#将数据第4列和第1列当作索引

# df

# user\_idauction\_idcat\_idcat1propertybuy\_mountday

# 078629554441098319944500148665002252021458:86755362;13023209:3593274;10984217:21985...220140919

# 153211045717916191097500119932821458:11399317;1628862:3251296;21475:137325;16...120131011

# 224901372521896936223500124615001481521458:30992;1628665:92012;1628665:3233938;1628...120131011

# 391705600712515996043500188315001481521458:15841995;21956:3494076;27000458:59723383...220141023

# 444406917320487688075500136365000816821458:30992;13658074:3323064;1628665:3233941;1...120141103

#对购买日期进行转换(to\_datetime)

df['buy\_date']=pd.to\_datetime(df['day'],format='%Y%m%d',errors='coerce')#format='%Y%m%d'形如20140919的年月日转换为时间戳buy\_date(加入errors防止报错)

# df.dtypes

# user\_id object

# auction\_id int64

# cat\_id int64

# cat1 int64

# property object

# buy\_mount int64

# day int64

# buy\_date datetime64[ns]

# dtype: object

df['buy\_date'].dt.year.head(3)#year,month,days是时间戳的dt的属性 可以提取对应年月日,必须为timestamp才可以，具体的时间点

# 0 2014

# 1 2013

# 2 2013

# Name: buy\_date, dtype: int64

# 对时间进行计算(时间差由天数，小时和毫秒构成)

df['diff\_day'] = pd.datetime.now() - df['buy\_date']#pd.datetime.now()可以获取当前时间

# df['diff\_day']

# 0 2874 days 09:26:35.630226

# 1 3217 days 09:26:35.630226

# 2 3217 days 09:26:35.630226

# 3 2840 days 09:26:35.630226

# 4 2829 days 09:26:35.630226

# 5 2829 days 09:26:35.630226

# 6 3520 days 09:26:35.630226

# 7 3520 days 09:26:35.630226

# 8 3561 days 09:26:35.630226

# 9 3561 days 09:26:35.630226

# 10 3561 days 09:26:35.630226

# 11 3539 days 09:26:35.630226

# 12 3539 days 09:26:35.630226

# 13 3539 days 09:26:35.630226

# timedelta可以理解为时间差类型

df['diff\_day'].dt.days#提取天数

# 0 2874

# 1 3217

# 2 3217

# 3 2840

df['diff\_day'].dt.seconds# 提取秒数

# 0 34061

# 1 34061

# 2 34061

# 3 34061

# 4 34061

df['diff\_day'].dt.microseconds# 提取毫秒数

# 0 953995

# 1 953995

# 2 953995

# 3 953995

# 4 953995

# 可以使用pd.Timedelta进行转换

df['时间差'] = df['diff\_day']/pd.Timedelta('1 D') #转换为天数

# df['时间差']

# 0 2874.394236

# 1 3217.394236

# 2 3217.394236

# 3 2840.394236

# 4 2829.394236

df['时间差'] = df['diff\_day']/pd.Timedelta('1 M') #转换为分钟

# df['时间差']

# 0 4.139128e+06

# 1 4.633048e+06

# 2 4.633048e+06

# 3 4.090168e+06

# 4 4.074328e+06

# 5 4.074328e+06

df['时间差'].round(decimals=3)# 显示为3小数

# 0 4139127.699

# 1 4633047.699

# 2 4633047.699

# 3 4090167.699

df['diff\_day'].astype('timedelta64[D]')# 这种方式也可以，M 代表月份数,D代表天数,Y代表年份

# 0 2874.0

# 1 3217.0

# 2 3217.0

# 3 2840.0

**数据集**

链接：<https://pan.baidu.com/s/11JdQriEYwFW6ySrHuKDuiw>提取码：1111

**字符串数据处理**

•

Pandas中提供了字符串的函数,但只能对字符型变量进行使用

•

通过str方法访问相关属性

•

可以使用字符串的相关方法进行数据处理



摩托车的销售情况

•

Condition：摩托车新旧情况（new：新的 和used：使用过的）

•

Condition\_Desc：对当前状况的描述

•

Price：价格

•

Location：发获地址

•

Model\_Year：购买年份

•

Mileage：里程

•

Exterior\_Color：车的颜色

•

Make：制造商（牌子）

•

Warranty：保修

•

Model：类型

•

Sub\_Model：车辆类型

•

Type：种类

•

Vehicle\_Title：车辆主题

•

OBO：车辆仪表盘

•

Watch\_Count：表数

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

df1 = pd.read\_csv('MotorcycleData.csv',encoding='gbk')#摩托车信息

# df1

# ConditionCondition\_DescPriceLocationModel\_YearMileageExterior\_ColorMakeWarrantyModel...Vehicle\_TitleOBOFeedback\_PercWatch\_CountN\_ReviewsSeller\_StatusVehicle\_TileAuctionBuy\_NowBid\_Count

# 0Usedmint!!! very low miles$11,412McHenry, Illinois, United States2013.016,000BlackHarley-DavidsonUnspecifiedTouring...NaNFALSE8.1NaN2427Private SellerClearTrueFALSE28.0

# 1UsedPerfect condition$17,200Fort Recovery, Ohio, United States2016.060BlackHarley-DavidsonVehicle has an existing warrantyTouring...NaNFALSE10017657Private SellerClearTrueTRUE0.0

# 2UsedNaN$3,872Chicago, Illinois, United States1970.025,763Silver/BlueBMWVehicle does NOT have an existing warrantyR-Series...NaNFALSE100NaN136NaNClearTrueFALSE26.0

# 3UsedCLEAN TITLE READY TO RIDE HOME$6,575Green Bay, Wisconsin, United States2009.033,142RedHarley-DavidsonNaNTouring...NaNFALSE100NaN2920DealerClearTrueFALSE11.0

df1['Price'].str[0:4]# 字符串切片

0 $11,

1 $17,

2 $3,8

3 $6,5

4 $10,

df1['价格'] = df1['Price'].str.strip('$')#删除前导字符串和后置空格

#df1['价格']

# 0 11,412

# 1 17,200

# 2 3,872

# 3 6,575

# 4 10,000

# 5 1,500

# 6 24,900

# 7 1,400

df1['价格'] = df1['价格'].str.replace(',','.')#字符串替换str.replace('被替换字符串','用来替换的字符串')

# df1['价格']

# 0 11.412

# 1 17.200

# 2 3.872

# 3 6.575

# 4 10.000

# 5 1.500

df1['价格'].astype(float)#将价格转换为float

# 0 11.412

# 1 17.200

# 2 3.872

# 3 6.575

# 4 10.000

# 5 1.500

df1.Model.head(5)

df1['Location'].str.split(',') # 使用字符串分割,用于对文本的处理

# 0 [McHenry, Illinois, United States]

# 1 [Fort Recovery, Ohio, United States]

# 2 [Chicago, Illinois, United States]

# 3 [Green Bay, Wisconsin, United States]

df1['Mileage'].str.len() #计算字符串长度

# 0 6.0

# 1 2.0

# 2 6.0

# 3 6.0

#正则表达式

df1['Exterior\_Color'].str.extract(r'B(.\*)k', expand=False)#提取需要文本

# 0 lac

# 1 lac

# 2 NaN

# 3 NaN

# 4 NaN

df1.replace({'Exterior\_Color': r'Bl'}, {'Exterior\_Color': 'a'}, regex=True)#'Exterior\_Color列符合Bl规则的都替换为a：

# ConditionCondition\_DescPriceLocationModel\_YearMileageExterior\_ColorMakeWarrantyModel...Vehicle\_TitleOBOFeedback\_PercWatch\_CountN\_ReviewsSeller\_StatusVehicle\_TileAuctionBuy\_NowBid\_Count

# 0Usedmint!!! very low miles$11,412McHenry, Illinois, United States2013.016,000aackHarley-DavidsonUnspecifiedTouring...NaNFALSE8.1NaN2427Private SellerClearTrueFALSE28.0

# 1UsedPerfect condition$17,200Fort Recovery, Ohio, United States2016.060aackHarley-DavidsonVehicle has an existing warrantyTouring...NaNFALSE

**数据集**

链接：<https://pan.baidu.com/s/1gDWAoxbJOBk6i41KwKrvgQ>提取码：1111

**高阶函数数据处理**

1.

在dataframe中使用apply方法，调用自定义函数对数据进行处理

2.

函数apply, axis=0表示对行进行操作,axis=1表示对列进行操作

3.

可以使用astype函数对数据进行转换

4.

可以使用map函数进行数据转换

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

df2 = pd.read\_csv('sam\_tianchi\_mum\_baby.csv',encoding = 'utf-8',dtype =str)#婴儿信息

# df2

# user\_idbirthdaygender

# 02757201303111

# 1415971201211110

# 21372572201201301

# 310339332201109100

# 410642245201302130

# 510923201201108301

def f(x):

if '0' in x:

return '女'

elif '1' in x :

return '男'

else:

return '未知'

# 0代表女，1代表男，2代表未知

df2['性别'] = df2['gender'].apply(f)#调用f方法

# df2['性别']

# 0 男

# 1 女

# 2 男

# 3 女

# 4 女

# 5 男

# 6 男

#使用map函数

df2['性别'] = df2['gender'].map({'0':'女','1':'男','2':'未知'})#效果和f方法一样

# df2['性别']

# 0 男

# 1 女

# 2 男

# 3 女

df2['user\_id'].apply(lambda x: x.replace(x[1:3],'\*\*')) #结合lambda替换

# 0 2\*\*7

# 1 4\*\*971

# 2 1\*\*2572

# 3 1\*\*39332

# 4 1\*\*42245

df2['birthday'].apply(lambda x: str(x)[0:4]) #提取年份

# 0 2013

# 1 2012

# 2 2012

# 3 2011

# 4 2013

**数据集**

链接：<https://pan.baidu.com/s/1hw8WzKohCKeShOxSQvlIVA>提取码：1111