Pandas cheat sheet

2019年4月14日 23:46

1. df.info()
2. df.tail(3)
3. #清除city字段中的字符空格
   df['city']=df['city'].map(str.strip)
4. city列大小写转换
   df['city']=df['city'].str.lower()
5. df.dtypees()
6. #更改数据格式
   df['price'].astype('int')
7. #更改列名称
   df.rename(columns={'category': 'category-size'})
8. #删除后出现的重复值
   df['city'].drop_duplicates()
9. #数据替换
   df['city'].replace('sh', 'shanghai')
10. #设置索引列
    df_inner.set_index('id')
11. #按特定列的值排序
    df_inner.sort_values(by=['age'])
12. #如果price列的值>3000，group列显示high，否则显示low
    df_inner['group'] = np.where(df_inner['price'] > 3000,'high','low')
13. #对复合多个条件的数据进行分组标记
    df_inner.loc[(df_inner['city'] == 'beijing') & (df_inner['price']
    >= 4000), 'sign']=1
14. #对category字段的值依次进行分列，并创建数据表，索引值为df_inner的索引列，列名称为category和size
    pd.DataFrame((x.split('-') for x in df_inner['category']),index=df_inner.index,columns=['category','size'])
15. #按索引提取单行的数值
    df_inner.loc[3]

    #.loc只按第几行，与index无关
16. #使用iloc按位置单独提取数据
    df_inner.iloc[[0,2,5],[4,5]]
    使用iloc按位置区域提取数据
    df_inner.iloc[:3,:2]
17. #使用ix按索引标签和位置混合提取数据
    df_inner.ix[:'2013-01-03',:4]
18. #先判断city列里是否包含beijing和shanghai，然后将复合条件的数据提取出来。
    df_inner.loc[df_inner['city'].isin(['beijing','shanghai'])]
19. #使用query函数进行筛选
    df_inner.query('city == ["beijing", "shanghai"]')
20. #对筛选后的数据按city列进行计数
    df_inner.loc[(df_inner['city'] != 'beijing'), ['id','city','age',
    'category','gender']].sort(['id']).city.count()
21. #对特定的ID列进行计数汇总
    df_inner.groupby('city')['id'].count()
    #对两个字段进行汇总计数
    df_inner.groupby(['city','size'])['id'].count()
22. #对city字段进行汇总并计算price的合计和均值。
    df_inner.groupby('city')['price'].agg([len,np.sum, np.mean])

	len	sum	mean
city			
beijing	2	5632	2816
guangzhou	1	2133	2133
shanghai	2	6598	3299
shenzhen	1	5433	5433

23. #数据透视表
pd.pivot_table(df_inner,index=["city"],values=["price"],columns=[
"size"],aggfunc=[len,np.sum],fill_value=0,margins=True)
24. #简单的数据采样
df_inner.sample(n=3)
25. #采样后不放回
df_inner.sample(n=6, replace=False)
26. 手动设置采样权重
weights = [0, 0, 0, 0, 0.5, 0.5]
df_inner.sample(n=2, weights=weights)
27. ★ #数据表描述性统计
df_inner.describe().round(2).T
28. #标准差
df_inner['price'].std()
1523.3516556155596
29. #两个字段间的协方差
df_inner['price'].cov(df_inner['m-point'])
17263.200000000001
30. #数据表中所有字段间的协方差
df_inner.cov()
31. #相关性分析
df_inner['price'].corr(df_inner['m-point'])
0.77466555617085264
32. #数据表相关性分析
df_inner.corr()
33. #输出到Excel格式
df_inner.to_Excel('Excel_to_Python.xlsx', sheet_name='bluewhale_c
c')
34. #对用户年龄进行分组
bins = [0, 18, 30, 50, 131]
group_age = ['少年', '青年', '中年', '老年']
cb['group_age'] = pd.cut(cb['age'], bins, labels=group_age)