从Excel到Python数据分析进阶指南

一、生成数据表

1. 导入数据表：df=pd.DataFrame(pd.read_csv('name.csv',header=1))

二、数据表检查

1. df.**shape**：查看数据表的维度
2. df.**info()**：查看数据表的整体信息
3. df.**dtypes**：可以一次性查看数据表中所 有数据的格式，也可以指定一列来单独查看（`df['B'].dtype`）
4. df.**isnull()**：检验空值的函数，返回的结果是逻辑值，包含空 值返回True，不包含则返回False
5. df['city'].**unique()**：查看唯一值的函数，只能对数据表中的特定列进行检查
6. df.**values**：查看数据表中的数值。以数组的形式 返回，不包含表头信息。
7. df.**columns**：查看列名称
8. df.**head(3)**：用来查看数据表中的前N行数据，默认head()显示前10 行，自己设置参数值来确定查看的行数。
9. df.**tail(3)**：用来查看数据表中后N行的数据，默认 tail()显示后10行，可以自己设置参数值来确定查看的行数。

三、数据表清洗

1. df.**dropna**(how='any')：删除数据表中含有空值的行
2. df.**fillna**(value=0)：使用数字0填充数据表中空值
3. df['price'].**fillna**(df['price'].mean())：使用price均值对NA进行填充
4. df['city']=df['city'].**map(str.strip)**：清除city字段中的字符空格
5. df['city']=df['city'].**str.lower()**：所有字母转换为小写，大写**upper**
6. df['price'].**astype**('int')：用来更改数据格式
7. df.**rename**(columns={'category': 'category-size'})：更改列名称的函数，我们将来数据表中的category列更

   改为category-size
8. df['city'].**drop_duplicates**()：删除重复值，默认删 除后出现的重复值，增加keep='last'参数后将删除

   最先出现的重复值，保留最后的值
9. df['city'].**replace**('sh', 'shanghai')：数据替换

四、数据预处理

1. df_inner=pd.**merge**(df,df1,how='inner')：对两个数据表进行合并
    1. inner：两个数据表中共有的数据匹配到一起生成新的数据表

    2. outer：外连接是保留两个表的所有信息，拼接的时候遇到标签不能对齐的部分，用NAN进行填充
    3. left：左连接是保留所有左表的信息，把右表中主键与左表一致的信息拼接进来，标签不能对齐的部分用NAN进行填充
    4. right：右连接是保留所有右表的信息，把左表中主键与左表一致的信息拼接进来，标签不能对齐的部分用NAN进行填充
2. df_inner.**set_index**('id')：设置索引列
3. df_inner.**reset_index**()：重设索引
4. df_inner.**sort_values**(by=['age'])：按特定列的值排序
5. df_inner.**sort_index**()：按索引列排序
6. **Where**：函数用来对数据进行判断和分组

   df_inner['group'] = np.where(df_inner['price'] > 3000,'high','low ')：

   如果price列的值>3000，group列显示high，否则显示low

五、数据提取

1. **Loc**：函数按数据表的索引标签进行提取（提取行）
2. **iloc**：函数按位置对数据表中的数据进行提取
3. **ix**：是loc和iloc的混合，既能按索引标签提取，也能按位置进行数 据提取
4. **isin**：按指定条件对数据进行提取。`df_inner['city'].isin(['beijing'])`

   `df_inner.loc[df_inner['city'].isin(['beijing','shanghai'])]`

   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。

六、数据筛选

&：与

|：或

！=：非

七、数据汇总

1. **groupby**：是进行分类汇总的函数，按列名称出 现的顺序进行分组。常见的是计数和 求和两种

   `df_inner.groupby('city').count()`：对所有列进行计数汇总

   `df_inner.groupby('city')['id'].count()`：对特定的ID列进行计数汇总

   `df_inner.groupby(['city','size'])['id'].count()`：对两个字段进行汇总计数
2. **pivot_table**：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)`

   设定city为行字段，size为列字 段，price为值字段。分别计算price的数量和金额并且按行与列进行汇总。

八、数据统计

1. **Sample**：是进行数据采样的函数，设置n的数量就可以了。函数自 动返回参与的结果。
2. **Describe**：对数据进行描述统计。自动生成数据的数量，均 值，标准差等数据

   `df_inner.describe().round(2).T`

   round函数设置结果显示的小数位。并对结果数据进行转置
3. **Std**：函数用来结算特定数据列的标准差，`df_inner['price'].std()`
4. **cov**：函数计算两个字段或数据表中各字段间的协方差，

   `df_inner['price'].cov(df_inner['m-point'])`

   `df_inner.cov()`：数据表中所有字段间的协方差
5. **corr**：函数完成相关分析的操作，并返回相关系数

九、数据输出

df_inner.to_csv('Excel_to_Python.csv')：输出到CSV格式