总结100个Pandas中序列的实用函数

在分享《Pandas模块，我觉得掌握这些就够用了！》后有很多读者朋友给我私信，希望分享一篇关于Pandas模块中序列的各种常有函数的使用。经过一段时间的整理，本期将分享我认为比较常规的100个实用函数，这些函数大致可以分为六类，分别是统计汇总函数、数据清洗函数、数据筛选、绘图与元素级运算函数、时间序列函数和其他函数。

数据分析过程中，必然要做一些数据的统计汇总工作，那么对于这一块的数据运算有哪些可用的函数可以帮助到我们呢？具体看如下几张表。

函数	含义	说明
s.min	计算最小值	s指代Pandas中的序列对象
s.max	计算最大值	
s.sum	求和	
s.mean	计算平均值	
s.count	计数	统计非缺失元素的个数
s.size	计数	统计所有元素的个数
s.median	计算中位数	
s.var	计算方差	
s.std	计算标准差	

函数	含义	说明
s.quantile	计算任意分位数	
s.cov	计算协方差	
s.corr	计算相关系数	
s.skew	计算偏度	
s.kurt	计算峰度	
s.mode	计算众数	
s.describe	描述性统计	一次性返回多个统计结果
s.groupby	分组	
s.aggregate	聚合运算	可以自定义统计函数

```
import pandas as pd
import numpy as np
x = pd.Series(np.random.normal(2,3,1000))
y = 3*x + 10 + pd.Series(np.random.normal(1,2,1000))

# 计算x与y的相关系数
print(x.corr(y))

# 计算y的偏度
print(y.skew())

# 计算y的统计描述值
print(x.describe())

z = pd.Series(['A','B','C']).sample(n = 1000, replace = Tru
# 重新修改z的行索引
z.index = range(1000)
# 按照z分组，统计y的组内平均值
y.groupby(by = z).aggregate(np.mean)
```

```
0.975684016624
0.0469892750818
count    1000.000000
mean        1.966565
std         2.895306
min        -6.874026
25%         0.018639
50%         1.995137
75%         3.749172
max        10.894609
dtype: float64

A    17.165736
B    16.313331
C    17.349420
dtype: float64
```

函数	含义	说明
s.argmin	寻找最小值所在位置	
s.argmax	寻找最大值所在位置	
s.any	等价于逻辑或	
s.all	等价于逻辑与	
s.value_counts	频次统计	
s.cumsum()	运算累计和	
s.cumprod	运算累计积	
s.pct_change	运算比率	后一个元素与前一个元素的比率

```
# 统计z中个元素的频次
print(z.value_counts())

a = pd.Series([1,5,10,15,25,30])
# 计算a中各元素的累计百分比
print(a.cumsum() / a.cumsum()[a.size - 1])
```

```
C    349
B    340
A    311
dtype: int64
0    0.011628
1    0.069767
2    0.186047
3    0.360465
4    0.651163
5    1.000000
dtype: float64
```

同样，数据清洗工作也是必不可少的工作，在如下表格中罗列了常有的数据清洗的函数。

函数	含义	说明
s.duplicated	判断序列元素是否重复	
s.drop_duplicates	删除重复值	
s.hasnans	判断序列是否存在缺失	仅返回True或False
s.isnull	判断序列元素是否为缺失	返回与序列长度一样的bool值
s.notnull	判断序列元素是否不为缺失	返回与序列长度一样的bool值
s.dropna	删除缺失值	
s.fillna	缺失值填充	使用缺失值的前一个元素填充
s.ffill	前向填充缺失值	使用缺失值的后一个元素填充
s.ffill	后向填充缺失值	

```
x = pd.Series([10,13,np.nan,17,28,19,33,np.nan,27])
#检验序列中是否存在缺失值
print(x.hasnans)

# 将缺失值填充为平均值
print(x.fillna(value = x.mean()))

# 前向填充缺失值
print(x.ffill())
```

```
True
0    10.0
1    13.0
2    21.0
3    17.0
4    28.0
5    19.0
6    33.0
7    21.0
8    27.0
dtype: float64
```

```
0    10.0
1    13.0
2    13.0
3    17.0
4    28.0
5    19.0
6    33.0
7    33.0
8    27.0
dtype: float64
```

函数	含义	说明
s.dtypes	检查数据类型	
s.astype	类型强制转换	
pd.to_datetime	转日期时间型	
s.factorize	因子化转换	
s.sample	抽样	
s.where	基于条件判断的值替换	
s.replace	按值替换	不可使用正则
s.str.replace	按值替换	可使用正则，.str不能少
s.str.split.str	字符分割	，.str不能少

```
income = pd.Series(['12500元','8000元','8500元','15000元','9
# 将收入转换为整型
print(income.str[:-1].astype(int))

gender = pd.Series(['男','女','女','女','男','女'])
# 性别因子化处理
print(gender.factorize())

house = pd.Series(['大宁金茂府 | 3室2厅 | 158.32平米 | 南 | 精
                   '昌里花园 | 2室2厅 | 104.73平米 | 南 | 精装
                   '纺大小区 | 3室1厅 | 68.38平米 | 南 | 简装'
# 取出二手房的面积，并转换为浮点型
house.str.split('|').str[2].str.strip().str[:-2].astype(flo
```

```
0    12500
1     8000
2     8500
3    15000
4     9000
dtype: int32
(array([0, 1, 1, 1, 0, 1], dtype=int64), Index(['男', '女'], dtype='object'))

0    158.32
1    104.73
2     68.38
dtype: float64
```

数据分析中如需对变量中的数值做子集筛选时，可以巧妙的使用下表中的几个函数，其中部分函数既可以使用在序列身上，也基本可以使用在数据框对象中。

函数	含义	说明
s.isin	成员关系判断	
s.between	区间判断	
s.loc	条件判断	可使用在数据框中
s.iloc	索引判断	可使用在数据框中
s.compress	条件判断	
s.nlargest	搜寻最大的n个元素	
s.nsmallest	搜寻最小的n个元素	
s.str.findall	子串查询	可使用正则

```
np.random.seed(1234)
x = pd.Series(np.random.randint(10,20,10))

# 筛选出16以上的元素
print(x.loc[x > 16])

print(x.compress(x > 16))

# 筛选出13~16之间的元素
```

```
print(x[x.between(13,16)])

# 取出最大的三个元素
print(x.nlargest(3))

y = pd.Series(['ID:1 name:张三 age:24 income:13500',
               'ID:2 name:李四 age:27 income:25000',
               'ID:3 name:王二 age:21 income:8000'])
# 取出年龄，并转换为整数
print(y.str.findall('age:(\d+)').str[0].astype(int))
```

```
4    18
5    19
7    17
8    19
dtype: int32
```

```
4    18
5    19
7    17
8    19
dtype: int32
```

```
0    13
1    16
2    15
3    14
9    16
dtype: int32
```

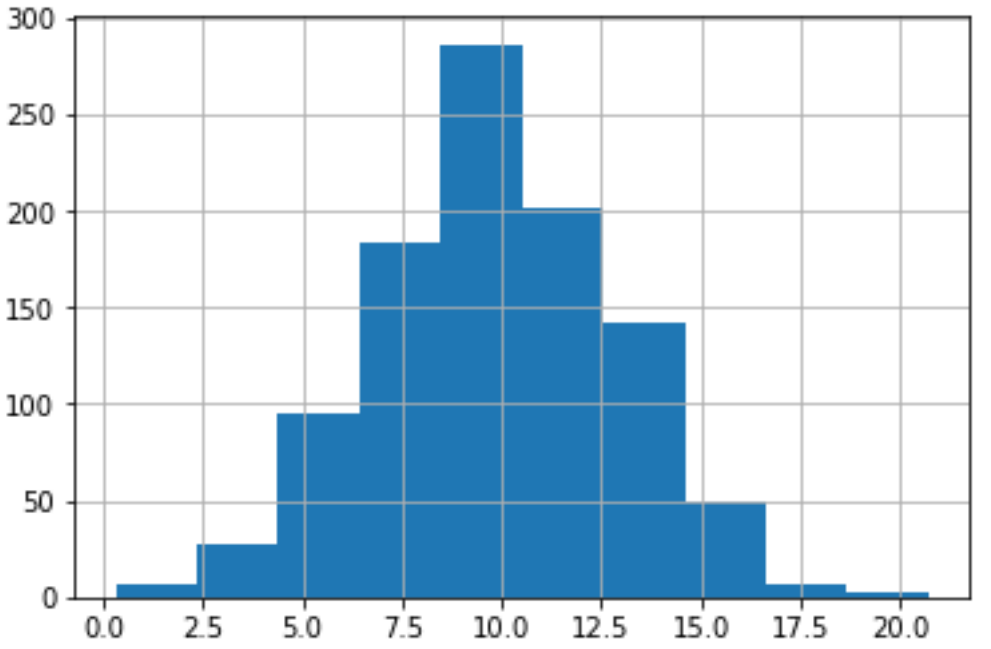
```
5    19
8    19
4    18
dtype: int32
```

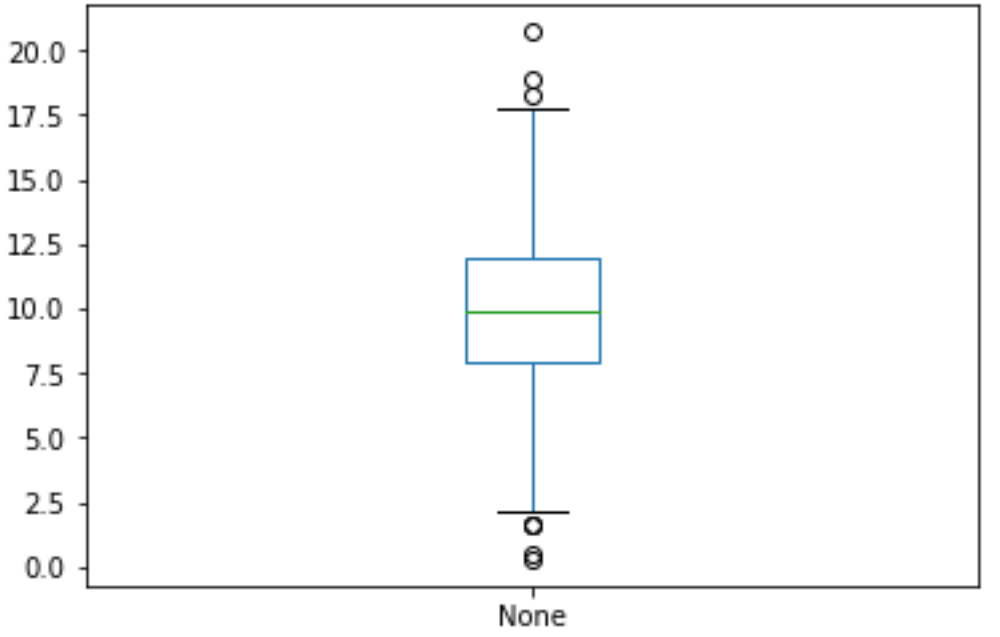
```
0    24
1    27
2    21
dtype: int32
```

```
np.random.seed(123)
import matplotlib.pyplot as plt
x = pd.Series(np.random.normal(10,3,1000))
# 绘制x直方图
x.hist()
# 显示图形
plt.show()

# 绘制x的箱线图
x.plot(kind='box')
plt.show()
```

```
installs = pd.Series(['1280万','6.7亿','2488万','1892万','98
# 将安装量统一更改为“万”的单位
def transform(x):
    if x.find('亿') != -1:
        res = float(x[:-1])*10000
    elif x.find('万') != -1:
        res = float(x[:-1])
    else:
        res = float(x)/10000
    return res
installs.apply(transform)
```






```
0      1280.0000
1     67000.0000
2      2488.0000
3      1892.0000
4         0.9877
5      9877.0000
6     12000.0000
dtype: float64
```

函数	含义	说明
s.dt.date	抽取出日期值	s为日期时间型序列，年-月-日
s.dt.time	抽取出时间	时:分:秒
s.dt.year	抽取出年	
s.dt.month	抽取出月	
s.dt.day	抽取出日	
s.dt.hour	抽取出时	