

## 11 | 数据科学家80%时间都花费在了这些清洗任务上？

2019-01-07 陈旸



讲述：陈旸

时长 09:45 大小 8.94M



我们在上一节中讲了数据采集，以及相关的工具使用，但做完数据采集就可以直接进行挖掘了吗？肯定不是的。

就拿做饭打个比方吧，对于很多人来说，热油下锅、掌勺翻炒一定是做饭中最过瘾的环节，但实际上炒菜这个过程只占做饭时间的 20%，剩下 80% 的时间都是在做准备，比如买菜、择菜、洗菜等等。

在数据挖掘中，数据清洗就是这样的前期准备工作。对于数据科学家来说，我们会遇到各种各样的数据，在分析前，要投入大量的时间和精力把数据“**整理裁剪**”成自己想要或需要的样子。

为什么呢？因为我们采集到的数据往往有很多问题。

我们先看一个例子，假设老板给你以下的数据，让你做数据分析，你看到这个数据后有什么感觉呢？

	0	1	2	3	4	5	6	7	8	9
0	1.0	Mickéy Mousé	56.0	70kgs	72	69	71	-	-	-
1	2.0	Donald Duck	34.0	154.89lbs	-	-	-	85	84	76
2	3.0	Mini Mouse	16.0	NaN	-	-	-	65	69	72
3	4.0	Scrooge McDuck	NaN	78kgs	78	79	72	-	-	-
4	5.0	Pink Panther	54.0	198.658lbs	-	-	-	69	NaN	75
5	6.0	Huey McDuck	52.0	189lbs	-	-	-	68	75	72
6	7.0	Dewey McDuck	19.0	56kgs	-	-	-	71	78	75
7	8.0	Scööpy Doo	32.0	78kgs	78	76	75	-	-	-
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	9.0	Huey McDuck	52.0	189lbs	-	-	-	68	75	72
10	10.0	Louie McDuck	12.0	45kgs	-	-	-	92	95	87

你刚看到这些数据可能会比较懵，因为这些数据缺少标注。

我们在收集整理数据的时候，一定要对数据做标注，数据表头很重要。比如这份数据表，就缺少列名的标注，这样一来我们就不知道每列数据所代表的含义，无法从业务中理解这些数值的作用，以及这些数值是否正确。但在实际工作中，也可能像这个案例一样，数据是缺少标注的。

我简单解释下这些数据代表的含义。

这是一家服装店统计的会员数据。最上面的一行是列坐标，最左侧一列是行坐标。

列坐标中，第 0 列代表的是序号，第 1 列代表的会员的姓名，第 2 列代表年龄，第 3 列代表体重，第 4~6 列代表男性会员的三围尺寸，第 7~9 列代表女性会员的三围尺寸。

了解含义以后，我们再看下中间部分具体的数据，你可能会想，这些数据怎么这么“脏乱差”啊，有很多值是空的（NaN），还有空行的情况。

是的，这还仅仅是一家商店的部分会员数据，我们一眼看过去就能发现一些问题。日常工作中的数据业务会复杂很多，通常我们要统计更多的数据维度，比如 100 个指标，数据量

通常都是超过 TB、EB 级别的，所以整个数据分析的处理难度是呈指数级增加的。这个时候，仅仅通过肉眼就很难找到问题所在了。

我举了这样一个简单的例子，带你理解在数据分析之前为什么要有数据清洗这个重要的准备工作。有经验的数据分析师都知道，**好的数据分析师必定是一名数据清洗高手，要知道在整个数据分析过程中，不论是在时间还是功夫上，数据清洗大概都占到了 80%。**

## 数据质量的准则

在上面这个服装店会员数据的案例中，一看到这些数据，你肯定能发现几个问题。你是不是想知道，有没有一些准则来规范这些数据的质量呢？

准则肯定是有的。不过如果数据存在七八种甚至更多的问题，我们很难将这些规则都记住。有研究说一个人的短期记忆，最多可以记住 7 条内容或信息，超过 7 条就记不住了。而数据清洗要解决的问题，远不止 7 条，我们万一漏掉一项该怎么办呢？有没有一种方法，我们既可以很方便地记住，又能保证我们的数据得到很好的清洗，提升数据质量呢？

在这里，我将数据清洗规则总结为以下 4 个关键点，统一起来叫“**完全合一**”，下面我来解释下。

1. **完整性**：单条数据是否存在空值，统计的字段是否完善。
2. **全面性**：观察某一列的全部数值，比如在 Excel 表中，我们选中一列，可以看到该列的平均值、最大值、最小值。我们可以通过常识来判断该列是否有问题，比如：数据定义、单位标识、数值本身。
3. **合法性**：数据的类型、内容、大小的合法性。比如数据中存在非 ASCII 字符，性别存在了未知，年龄超过了 150 岁等。
4. **唯一性**：数据是否存在重复记录，因为数据通常来自不同渠道的汇总，重复的情况是常见的。行数据、列数据都需要是唯一的，比如一个人不能重复记录多次，且一个人的体重也不能在列指标中重复记录多次。

在很多数据挖掘的教学中，数据准则通常会列出来 7~8 项，在这里我们归类成了“**完全合一**” 4 项准则，按照以上的原则，我们能解决数据清理中遇到的大部分问题，使得**数据标准、干净、连续**，为后续数据统计、数据挖掘做好准备。如果想要进一步优化数据质量，还需要在实际案例中灵活使用。

## 清洗数据，一一击破

了解了数据质量准则之后，我们针对上面服装店会员数据案例中的问题进行一一击破。

这里你就需要 Python 的 Pandas 工具了。这个工具我们之前介绍过。它是基于 NumPy 的工具，专门为解决数据分析任务而创建。Pandas 纳入了大量库，我们可以利用这些库高效地进行数据清理工作。

这里我补充说明一下，如果你对 Python 还不是很熟悉，但是很想从事数据挖掘、数据分析相关的工作，那么花一些时间和精力来学习一下 Python 是很有必要的。Python 拥有丰富的库，堪称数据挖掘利器。当然了，数据清洗的工具也还有很多，这里我们只是以 Pandas 为例，帮你应用数据清洗准则，带你更加直观地了解数据清洗到底是怎么回事儿。

下面，我们就依照“完全合一”的准则，使用 Pandas 来进行清洗。

### 1. 完整性

#### 问题 1：缺失值


在数据中有些年龄、体重数值是缺失的，这往往是因为数据量较大，在过程中，有些数值没有采集到。通常我们可以采用以下三种方法：

删除：删除数据缺失的记录；

均值：使用当前列的均值；

高频：使用当前列出现频率最高的数据。

比如我们想对 `df['Age']` 中缺失的数值用平均年龄进行填充，可以这样写：

 复制代码

```
1 df['Age'].fillna(df['Age'].mean(), inplace=True)
```


如果我们用最高频的数据进行填充，可以先通过 `value_counts` 获取 Age 字段最高频次 `age_maxf`，然后再对 Age 字段中缺失的数据用 `age_maxf` 进行填充：



```
1 age_maxf = train_features['Age'].value_counts().index[0]
2 train_features['Age'].fillna(age_maxf, inplace=True)
```

## 问题 2：空行

我们发现数据中有一个空行，除了 index 之外，全部的值都是 NaN。Pandas 的 `read_csv()` 并没有可选参数来忽略空行，这样，我们就需要在数据被读入之后再使用 `dropna()` 进行处理，删除空行。

 复制代码


```
1 # 删除全空的行
2 df.dropna(how='all', inplace=True)
```

## 2. 全面性

### 问题：列数据的单位不统一

观察 `weight` 列的数值，我们能发现 `weight` 列的单位不统一。有的单位是千克（`kgs`），有的单位是磅（`lbs`）。

这里我使用千克作为统一的度量单位，将磅（`lbs`）转化为千克（`kgs`）：


 复制代码

```
1 # 获取 weight 数据列中单位为 lbs 的数据
2 rows_with_lbs = df['weight'].str.contains('lbs').fillna(False)
3 print df[rows_with_lbs]
4 # 将 lbs 转换为 kgs, 2.2lbs=1kgs
5 for i, lbs_row in df[rows_with_lbs].iterrows():
6     # 截取从头开始到倒数第三个字符之前，即去掉 lbs。
7     weight = int(float(lbs_row['weight'][:-3])/2.2)
8     df.at[i, 'weight'] = '{}kgs'.format(weight)
```

## 3. 合理性

## 问题：非 ASCII 字符

我们可以看到在数据集中 Fristname 和 Lastname 有一些非 ASCII 的字符。我们可以采用删除或者替换的方式来解决非 ASCII 问题，这里我们使用删除方法：


 复制代码

```
1 # 删除非 ASCII 字符
2 df['first_name'].replace({r'^\x00-\x7F]+'}, regex=True, inplace=True)
3 df['last_name'].replace({r'^\x00-\x7F]+'}, regex=True, inplace=True)
```

## 4. 唯一性

### 问题 1：一列有多个参数


在数据中不难发现，姓名列（Name）包含了两个参数 Firtname 和 Lastname。为了达到数据整洁目的，我们将 Name 列拆分成 Firstname 和 Lastname 两个字段。我们使用 Python 的 split 方法，str.split(expand=True)，将列表拆成新的列，再将原来的 Name 列删除。

 复制代码

```
1 # 切分名字，删除源数据列
2 df[['first_name','last_name']] = df['name'].str.split(expand=True)
3 df.drop('name', axis=1, inplace=True)
```

### 问题 2：重复数据

我们校验一下数据中是否存在重复记录。如果存在重复记录，就使用 Pandas 提供的 drop\_duplicates() 来删除重复数据。

 复制代码

```
1 # 删除重复数据行
2 df.drop_duplicates(['first_name','last_name'],inplace=True)
```

这样，我们就将上面案例中的会员数据进行了清理，来看看清理之后的数据结果。怎么样？是不是又干净又标准？

FirstName	LastName	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218
Micky	Mous	56.0	70kgs	72	69	71			
Donald	Duck	34.0					85	84	76
Mini	Mouse	16.0					65	69	72
Scrooge	McDuck	36.3	78kgs	78	79	72			
Pink	Panther	54.0	90kgs				69	79	75
Huey	McDuck	52.0	85kgs				68	75	72
Dewey	McDuck	19.0	56kgs				71	78	75
Scoopy	Doo	32.0	78kgs	78	76	75			
Huey	McDuck	52.0					68	75	72
Louie	McDuck	12.0					92	95	87

## 养成数据审核的习惯

现在，你是不是能感受到数据问题不是小事，上面这个简单的例子里都有 6 处错误。所以我们常说，现实世界的数据是“肮脏的”，需要清洗。

第三方的数据要清洗，自有产品的数据，也需要数据清洗。比如美团自身做数据挖掘的时候，也需要去除爬虫抓取，作弊数据等。可以说**没有高质量的数据，就没有高质量的数据挖掘，而数据清洗是高质量数据的一道保障。**

当你从事这方面工作的时候，你会发现养成数据审核的习惯非常重要。而且越是优秀的数据挖掘人员，越会有“数据审核”的“职业病”。这就好比编辑非常在意文章中的错别字、语法一样。

数据的规范性，就像是你的作品一样，通过清洗之后，会变得非常干净、标准。当然了，这也是一门需要不断修炼的功夫。终有一天，你会进入这样一种境界：看一眼数据，差不多 7 秒钟的时间，就能知道这个数据是否存在问题。为了这一眼的功力，我们要做很多练习。

刚开始接触数据科学工作的时候，一定会觉得数据挖掘是件很酷、很有价值的事。确实如此，不过今天我还要告诉你，再酷炫的事也离不开基础性的工作，就像我们今天讲的数据清洗工作。对于这些基础性的工作，我们需要耐下性子，一个坑一个坑地去解决。

好了，最后我们来总结下今天的内容，你都收获了什么？

# 数据清理

一、意义：(数据科学家80%的时间都花在了数据清理上。)

没有好的数据清理，就没有好的数据挖掘。

二、数据质量准则：

“完全合一”

三、解决问题：

使用工具箱，利用Pandas

Tips:

我们做任何事情，准备工作都是基础且必要的，要养成数据审核的习惯。

学习完今天的内容后，给你留个小作业吧。下面是一个美食数据，如果你拿到下面的数据，按照我们今天讲的准则，你能找到几点问题？如果你来清洗这些数据，你打算怎样清洗呢？

food	ounces	animal
bacon	4.0	pig
pulled pork	3.0	pig
bacon	NaN	pig
Pastrami	6.0	cow
corned beef	7.5	cow
Bacon	8.0	pig
pastrami	-3.0	cow
honey ham	5.0	pig
nova lox	6.0	salmon



欢迎在留言区写下你的思考，如果你对今天“数据清洗”的内容还有疑问，也欢迎留言和我讨论。也欢迎点击“请朋友读”，把这篇文章分享给你的朋友或者同事。

 极客时间

# 数据分析实战 45 讲

## 即学即用的数据分析入门课



陈旻  
清华大学计算机博士

新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得转载

上一篇 10 | Python爬虫：如何自动化下载王祖贤海报？

下一篇 12 | 数据集成：这些大号一共20亿粉丝？

## 精选留言 (51)

 写留言



Hot H...

2019-01-07

 13

可以给个样例数据的链接吗？自己动手操作一下

展开 ▾



wonderland

2019-01-10

 6

一、首先按照所讲的数据质量准则，数据存在的问题有：

1. "完整性"问题：数据有缺失，在ounces列的第三行存在缺失值

解决办法：可以用该列的平均值来填充此缺失值

2. "全面性"问题：food列的值大小写不统一

解决办法：统一改为小写...

展开 ∨



geektime ...

2019-01-07

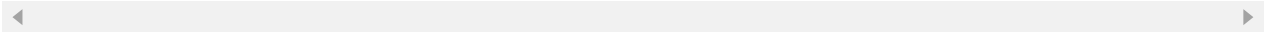
👍 6

这些东西，大家都一定要上手去实现一遍。最简单的就是，搞一个文本，把这些数据放进去，用Python读这个文本，转成dataframe，把老师讲的那些清洗相关的API都一个一个试一下，才会有体会，光看一遍真的没啥用的！

现在只是很少的几十条数据，等你真正去搞那些上亿的数据的时候，就知道核对数据是个多么复杂的事情了.....

展开 ∨

作者回复: 对的 一定要自己模拟操作下



nrvna

2019-01-09

👍 4

jupyter notebook , python3

```
import pandas as pd
df = pd.read_csv("D://Data_for_sci/food.csv")
df.index...
```

展开 ∨



上官

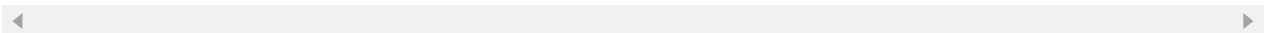
2019-01-08

👍 4

```
weight = int(float((lbs_row['weight'][:-3])/2.2)
```

老师好，这行代码中[:-3]的作用是什么啊？

作者回复: 截取从头开始到倒数第三个字符之前，即去掉lbs。





**third**

2019-02-05

👍 3

自己不知道有没有什么好的工具，所以就把图片上一个一个敲进去了。

数据.csv格式

链接：<https://pan.baidu.com/s/1jNnUpntrlxFSbmna3HtXw>

提取码：e9hc



**auroroa**

2019-01-07

👍 3

最大的问题是不是没把数据的来源和目的描述清楚？☺

展开 ▼



**桃园悠然在**

2019-01-07

👍 3

我的理解，不能对food列简单去重吧，而是规范ounces列数据后汇总或者保持原样，这可能使厨房食材消耗记录。数据清洗还是要结合完全合一+业务含义。



**Jbin**

2019-01-07

👍 3

练习题中：

- 1、food列中出现大小写不同的情况。根据实际，如果大小写不同的两个数据代表的产品不同，则不改变，否则统一改为小写
- 2、food列bacon出现了三次，但是有两次是有正确数据，不能通过food去重。
- 3、ounces列，去除空值行。根据实际数据来源以及分析目的，是否可能有负的情况，...

展开 ▼



**奋斗**

2019-01-07

👍 2

老师你好！我是爬虫新手，在为机器翻译提供语料，爬取完数据很头疼，文本数据里有很多问题，老师针对文本类的数据怎么处理好那，pandas适用吗？谢谢了



**Tommy**

2019-01-07

👍 2

脚本看不全啊

展开 ∨

---



晨星

2019-02-19

👍 1

```
import pandas as pd
"""利用Pandas清洗美食数据"""
```

```
# 读取csv文件
df = pd.read_csv("c.csv")...
```

展开 ∨

---



雨先生的晴...

2019-01-14

👍 1

老师 你好，按照您的方法我清理一下数据，有一些疑惑，希望能指正。  
在 ‘服装店统计的会员数据’ 例子 最终清洗截图中，最后截图中，  
No.1 为什么 Huey McDuck 出现了两次？  
No.2 为什么 Donald Duck 的体重数据 没有转化成Kgs?  
我试着按照例子，自己做了一边，发现以下代码，之修改成功了 Huey McDuck的体重...

展开 ∨

---



周飞

2019-01-12

👍 1

完整性：ounces 列数据中存在NAN  
全面性：food列数据中存在大小写不一致问题  
合法性：ounces列数据存在负值  
唯一性：food列数据存在重复  
# -\*- coding: utf-8 -\*-...

展开 ∨

---



北方

2019-01-11

👍 1

```
#!/usr/bin/env python
# -*- coding:utf8 -*-
# __author__ = '北方姆Q'
# __datetime__ = 2019/1/11 15:53
```

...

展开 ∨

---



王

2019-02-28



第三个规则，合值得是合理性还是合法性呢？

展开 ∨

---



一语中的

2019-02-26



#pd读取数据

```
df = pd.read_excel('testdata11.xlsx')
```

#1.完整性，ounces列NA值用平均值填充

```
df['ounces'].fillna(df['ounces'].mean(), inplace=True)
```

#2.全面性，统一food列大小写...

展开 ∨

---



徐薛彪

2019-02-24



## 首先样本数据集存在如下问题:

完整性: ounces列存在空值

全面性:

合理性: ounces列存在负值

唯一性: bacon美食存在多条记录，并且名称存在大小写，无法唯一区分一个食物信息...

展开 ∨

---



littlePerf...

2019-02-18



```
import pandas as pd
```

```
df = pd.read_excel("E:\data_analys_work/food_data.xlsx")
```

# 1. 完整性问题: 缺失值...

展开 ∨

---





草包雷

2019-02-18



```
import pandas as pd
df = pd.read_csv("c://leijin//food.csv")
df.index
print (df)
```

...

展开 ∨