

一个「神奇」的Python库，99%的人都爱！

法纳斯特 2022-04-06 14:00

以下文章来源于数据分析与统计学之美，作者黄伟呢



数据分析与统计学之美

免费领10w字"Python知识手册"，共400页，后台回复“十万”领取！

大家好，我是小F~

今天介绍Python中一个超级神奇的库，**99%人用过都喜欢它，剩下的1%没用过！**

在如今的大数据时代，数据的价值可想而知。有时候为了做测试，需要模拟真实的环境，但是又不能直接使用真实数据，就需要我们人为制造一些数据出来。

对比Excel，我还是觉得Python制造这样的“虚拟”数据，更省时、省力。

周末，突然想到了曾今做过的这个问题，这里为大家做个复盘吧！



需求：老板让模拟一批数据，用于项目实验，由于一些真实数据不能展示出来，我需要模拟一些数据，字段包括：[姓名](#)、[所在省份](#)、[详细地址](#)、[手机号](#)、[身份证号](#)、[出生年月](#)、[邮箱](#)等。

当然，这批数据肯定是需要你最终写入到Excel中，一次性交给老板的。那么，这样的需求，你会做吗？

模拟1w条数据写入Excel

在讲述基础之前，直接上实战，让大家体会一下，如何将生成的模拟数据，最终写入到Excel文件中。

```
from faker import Faker
import pandas as pd

fake = Faker(["zh_CN"])
Faker.seed(0)

def get_data():
    key_list = ["姓名", "详细地址", "所在省份", "手机号", "身份证号", "出生年月", "邮箱"]
    name = fake.name()
    address = fake.address()
    province = address[:3]
    number = fake.phone_number()
    id_card = fake.ssn()
    birth_date = id_card[6:14]
    email = fake.email()
    info_list = [name, address, province, number, id_card, birth_date, email]
    person_info = dict(zip(key_list, info_list))
```

```
return person_info

df = pd.DataFrame(columns=["姓名", "详细地址", "所在省份", "手机号", "身份证号", "出生年月", "邮箱"])

for i in range(10000):
    person_info = [get_data()]
    df1 = pd.DataFrame(person_info)
    df = pd.concat([df, df1])

df.to_excel("模拟数据.xlsx", index=None)
```

结果如下：

	A	B	C	D	E	F	G
	姓名	详细地址	所在省份	手机号	身份证号	出生年月	邮箱
1	廖婷婷	湖南省秀云县和平东莞街1座 594892	湖南省	13715781565	511501194810013878	19481001	fang80@gmail.com
2	程伟	贵州省兴城市南长武汉路f座 587148	贵州省	18518583989	411527197710011967	19771001	lihe@gmail.com
3	王秀兰	内蒙古自治区太原市清城龙路n座 294775	内蒙古	15791795330	331121193712061353	19371206	ping01@jietan.cn
4	王坤	陕西省张家港市清城昆明街n座 508691	陕西省	13631456208	130705196108259169	19610825	ming79@sr.cn
5	韦晨	贵州省欢县南湖余路y座 628071	贵州省	18608423759	520382197709235996	19770923	na61@93.cn
6	刘秀芳	甘肃省马林县兴山王街h座 689007	甘肃省	15270638120	520602196707056505	19670705	baiyong@xiuyingduan.cn
7	向萍	河北省桂珍市秀英侯街i座 112400	河北省	14748559097	511621193908257658	19390825	nayan@chao.cn
8	杨洁	贵州省长春市长寿黄街h座 230428	贵州省	15265461187	360302196101115172	19610111	cshe@min.cn
9	祝淑珍	内蒙古自治区拉萨市清河西安路C座 493615	内蒙古	15192635110	419001195605203171	19560520	juan03@hotmail.com
10	李桂芳	四川省银川市沙市温路F座 638930	四川省	18555082492	45102519690721471X	19690721	dengjun@ct.cn
11	陈瑞	新疆维吾尔自治区喀什县梁平柳州街w座 748967	新疆维	14593002489	360101198009221740	19800922	zqiao@yahoo.com
12	张阳	河南省华市和平惠州街z座 860976	河南省	13517200992	350629194701151850	19470115	htian@97.cn
13	蔡英	云南省佳市清城黄街w座 900740	云南省	15699331886	341821195801231261	19580123	fanping@gmail.com
14	赵建军	青海省丽娟市永川李街v座 655203	青海省	13692303104	522729197006100931	19700610	qiangfang@meng.cn
15	李利	贵州省桂兰县吉区银川路y座 423583	贵州省	14524560290	140828196803231266	19680323	lgong@yahoo.com
16	曾凯	青海省华县朝阳奕路n座 547121	青海省	13789276266	33011019871118758X	19871118	qiang17@juanli.cn
17	冯强	广东省即聊市房山汕尾街r座 714004	广东省	13096714543	141027195002200140	19500220	guiying83@gmail.com
18	赵桂兰	澳门特别行政区柳州市丰都楼路s座 259221	澳门特	15969728457	410401193612153779	19361215	dzeng@yahoo.com
19	李金凤	浙江省齐齐哈尔县东城兰州路x座 419044	浙江省	13894316756	350923198806176620	19880617	weilu@gmail.com
20	欧玉	云南省小红县沈河路街c座 412502	云南省	14549184567	511025199809198867	19980919	tlai@yahoo.com
21	杨刚	重庆市哈尔滨市牧野石家庄路M座 283735	重庆市	14515587540	13082119560922351X	19560922	juan34@liulong.cn
22	刘红梅	天津市城市静安王路D座 311131	天津市	18237916860	330201198705177325	19870517	min61@hotmail.com
23	郭强	天津市城市静安王路D座 311131	天津市	18237916860	330201198705177325	19870517	min61@hotmail.com

以上数据为模拟生成，如果雷同，纯属巧合！

Python库讲解

这么好用的Python库，究竟应该怎么使用呢？

我们直接使用下面的代码，可以完成这个库的安装。

```
pip install Faker -i https://pypi.tuna.tsinghua.edu.cn/simple/
```

使用之前，使用如下代码，导入这个库。

```
from faker import Faker
```

在讲述写入到Excel之前，我们先分布讲述一下，每个函数的用法。

1. 生成姓名

```
fake = Faker(locale='zh_CN')
name = fake.name()
name
```

结果如下：

```
In [63]: fake = Faker(locale='zh_CN')
         name = fake.name()
         name
Out[63]: '万宁'
```

2. 生成详细地址

```
address = fake.address()
address
```

结果如下：

```
In [64]: address = fake.address()
         address
Out[64]: '辽宁省哈尔滨县江北哈尔滨路u座 756619'
```

3. 生成所在省份

```
province = address[:3]
province
```

结果如下：

```
In [65]: province = address[:3]
         province
Out[65]: '辽宁省'
```

由于这个函数每次运行结果都不一样，所以我才用切片方式，生成省份。当然这里也有特定函数，生成省份。

```
fake.province()
```

结果如下：

```
In [66]: fake.province()
Out[66]: '湖南省'
```

4. 生成手机号

```
number = fake.phone_number()
number
```

结果如下：

```
In [67]: number = fake.phone_number()
         number
Out[67]: '13312128352'
```

5. 生成身份证号

```
id_card = fake.ssn()
id_card
```

结果如下：

```
In [68]: id_card = fake.ssn()
         id_card
Out[68]: '441523199201123281'
```

6. 生成出生年月

```
birth_date = id_card[6:14]
```

```
birth_date
```

结果如下：

```
In [69]: birth_date = id_card[6:14]
         birth_date
Out[69]: '19920112'
```

7. 生成邮箱

```
email = fake.email()
email
```

结果如下：

```
In [70]: email = fake.email()
         email
Out[70]: 'laiming@uc.cn'
```

补充

当然，`faker` 库不仅可以帮助我们生成上述信息，还有很多其它方法可用，这些方法分为以下几类：

- `address` 地址
- `person` 人物类：性别、姓名等
- `barcode` 条码类
- `color` 颜色类
- `company` 公司类：公司名、email、公司名前缀等
- `credit_card` 银行卡类：卡号、有效期、类型等
- `currency` 货币
- `date_time` 时间日期类：日期、年、月等
- `file` 文件类：文件名、文件类型、文件扩展名等
- `internet` 互联网类
- `job` 工作
- `lorem` 乱数假文
- `misc` 杂项类
- `phone_number` 手机号码类：手机号、运营商号段
- `python` python数据
- `profile` 人物描述信息：姓名、性别、地址、公司等
- `ssn` 社会安全码(身份证号码)
- `user_agent` 用户代理

关于这些方法的使用，我们直接参考faker的官网，用起来超方便。

faker.readthedocs.io/en/master/providers.html

1. address 地址

```
fake.country() # 国家
fake.city() # 城市
fake.city_suffix() # 城市的后缀,中文是：市或县
fake.address() # 地址
fake.street_address() # 街道
fake.street_name() # 街道名
fake.postcode() # 邮编
fake.latitude() # 纬度
fake.longitude() # 经度
```

2. person 人物

```
fake.name() # 姓名
fake.last_name() # 姓
fake.first_name() # 名
fake.name_male() # 男性姓名
fake.last_name_male() # 男性姓
fake.first_name_male() # 男性名
fake.name_female() # 女性姓名
```

3. color 颜色

```
fake.hex_color() # 16进制表示的颜色
fake.rgb_css_color() # css用的rgb色
fake.rgb_color() # 表示rgb色的字符串
fake.color_name() # 颜色名字
fake.safe_hex_color() #安全16进制色
fake.safe_color_name() # 安全颜色名字
```

4. company 公司

```
fake.company() # 公司名
fake.company_suffix() # 公司名后缀
```

5. credit_card 银行信用卡

```
fake.credit_card_number(card_type=None) # 卡号
fake.credit_card_provider(card_type=None) # 卡的提供者
fake.credit_card_security_code(card_type=None) # 卡的安全密码
fake.credit_card_expire() # 卡的有效期
fake.credit_card_full(card_type=None) # 完整卡信息
```

6. date_time 时间日期

```
fake.date_time(tzinfo=None) # 随机日期时间
fake.iso8601(tzinfo=None) # 以iso8601标准输出的日期
fake.date_time_this_month(before_now=True, after_now=False, tzinfo=None) # 本月的某个日期
fake.date_time_this_year(before_now=True, after_now=False, tzinfo=None) # 本年的某个日期
fake.date_time_this_decade(before_now=True, after_now=False, tzinfo=None) # 本年代内的一个日期
fake.date_time_this_century(before_now=True, after_now=False, tzinfo=None) # 本世纪一个日期
fake.date_time_between(start_date="-30y", end_date="now", tzinfo=None) # 两个时间间的一个随机时间
fake.timezone() # 时区
fake.time(pattern="%H:%M:%S") # 时间（可自定义格式）
fake.am_pm() # 随机上午下午
fake.month() # 随机月份
fake.month_name() # 随机月份名字
fake.year() # 随机年
fake.day_of_week() # 随机星期几
fake.day_of_month() # 随机月中某一天
fake.time_delta() # 随机时间延迟
fake.date_object() # 随机日期对象
fake.time_object() # 随机时间对象
fake.unix_time() # 随机unix时间（时间戳）
fake.date(pattern="%Y-%m-%d") # 随机日期（可自定义格式）
fake.date_time_ad(tzinfo=None) # 公元后随机日期
```

7. file 文件

```
fake.file_name(category="image", extension="png") # 文件名（指定文件类型和后缀名）
fake.file_name() # 随机生成各类型文件
fake.file_extension(category=None) # 文件后缀
fake.mime_type(category=None) # mime-type
```

8. internet 互联网

```
fake.ipv4(network=False) # ipv4地址
fake.ipv6(network=False) # ipv6地址
fake.uri_path(deep=None) # uri路径
fake.uri_extension() # uri扩展名
fake.uri() # uri
fake.url() # url
fake.image_url(width=None, height=None) # 图片url
fake.domain_word() # 域名主体
fake.domain_name() # 域名
fake.tld() # 域名后缀
fake.user_name() # 用户名
fake.user_agent() # UA
fake.mac_address() # MAC地址
fake.safe_email() # 安全邮箱
fake.free_email() # 免费邮箱
fake.company_email() # 公司邮箱
fake.email() # 邮箱
```

9. job 工作

```
fake.job()#工作职位
```

10. lorem 乱数假文

```
fake.text(max_nb_chars=200) # 随机生成一篇文章
fake.word() # 随机单词
fake.words(nb=3) # 随机生成几个字
fake.sentence(nb_words=6, variable_nb_words=True) # 随机生成一个句子
fake.sentences(nb=3) # 随机生成几个句子
fake.paragraph(nb_sentences=3, variable_nb_sentences=True) # 随机生成一段文字(字符串)
fake.paragraphs(nb=3) # 随机生成成几段文字(列表)
```

11. phone_number 电话号码

```
fake.phone_number() # 手机号码
fake.phonenumber_prefix() # 运营商号段，手机号码前三位
```

12. ssn 社会安全码(身份证)

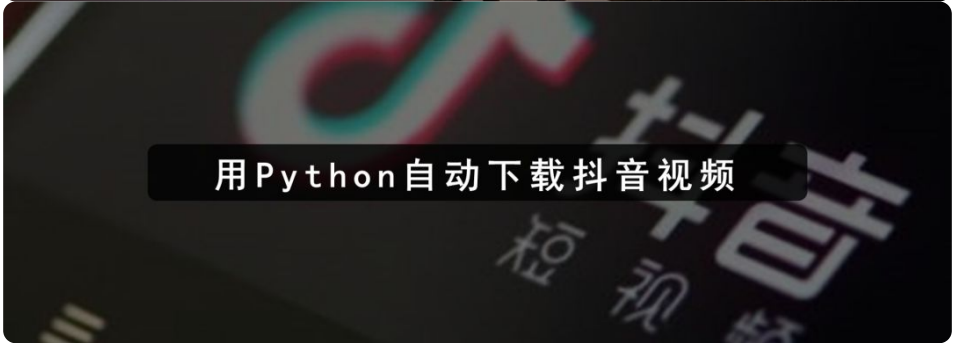
```
fake.ssn() # 随机生成身份证号(18位)
```

13. user_agent 用户代理

```
fake.user_agent()
```

万水千山总是情，点个 赞 行不行。

推荐阅读



... END ...



Python | 学习 | 生活

每一次打开都有收获



扫描二维码关注

喜欢此内容的人还喜欢

python asyncio 异步 I/O - 实现并发http请求(asyncio + aiohttp)
从零开始学自动化测试

python_mmdt:ssdeep、tlsh、vhash、mmdthash对比
汇编语言

linux工具——神奇的正则表达式
数据仓库践行者

