

[< Back to Data Analyst Nanodegree](#)

Explore US Bikeshare Data

审阅

代码审阅

HISTORY

Requires Changes

还需满足 4 个要求 变化

周末好~恭喜你完成了项目的第一次提交，这是一个很棒的开始，不过目前项目存在一些问题，无法正常运行，基于现有的代码给出了一些修改建议，期待你的下一次提交，相信你一定会做的更棒👍

代码质量

所有代码都可以正常工作，运行时不会产生错误。

```
city_data_filename = CITY_DATA[city]
KeyError: 'ChiCaGo'
```

如果用户没有按照预期输入的话，此时会报错。因为 `lower()` 方法只会将 `input_city` 的值转换为小写，但并不会改变 `input_city` 本身的内容，所以需要将转换为小写的变量赋值给 `input_city`，同理，`month` 和 `day` 也应该这样处理，目前这两个是没有做大小写容错处理的。比如 `city` 输入这部分可以写成这样，

```
while True:
    print('请输入你要选择的城市: "chicago, new york city, washington"')
    city = input().lower()
    if city in CITY_DATA.keys():
        break
print('-' * 40)
```

选择适当的数据类型（例如字符串，浮点型）和数据结构（例如列表，字典）来执行所需的分析任务。

循环和条件语句用于正确处理数据。

包用于执行高级任务，例如读写文件以及创建可视化文件。

使用函数，用于减少重复代码。

目前接收用户输入这部分代码有些重复，可以将这部分代码提取成一个函数，这样可以有效的减少重复代码，

```
def input_mod(input_print,error_print,enterable_list):  
    ret = input(input_print).lower()  
    while ret not in enterable_list:  
        ret = input(error_print).lower()  
    return ret
```

文档字符串，注释和变量名称使代码具有可读性。

脚本与问题

原始输入得以正确地征求和处理，以指导交互式问答体验; 添加意外输入时不会引发错误。

接收用户输入这部分没有进行容错处理，容错包括两部分，

1. 忽略用户输入的大小写
2. 处理用户的非法输入

同时在月份和星期的输入和过滤这部分，没有考虑 `all` 的情况，因为在接收输入的地方已经把星期和月份转换为数字了，如果用户输入 `all` 也会被转换为数字，那么在 `load_data` 这里，`if month != 'all'` 这个条件事实上是没有用的。所以这部分代码也需要修改。

描述性统计数据得以正确计算，并用于回答有关数据的问题。如果同学请求了原始数据，原始数据是依据同学的请求显示的。

统计数据这部分，

1. 没有计算热门旅程
2. 需要考虑华盛顿数据集不存在 `Gender` 和 `Birth Year` 这两个字段的情况，建议使用 `try except` 捕获异常

```
try:
    print(df['Gender'].value_counts())
except:
    print('no gender data')
```

☒ 重新提交

 下载项目



重新提交项目的最佳做法

Ben 与你分享修改和重新提交的 5 个有益的小贴士。

 [观看视频 \(3:01\)](#)

[返回 PATH](#)

[学员 FAQ](#)