



# Python 语言课程设计实习报告

姓 名	<u>曾康慧</u>	专 业	<u>自动化</u>
学 号	<u>20211003337</u>	班 学 号	<u>01D231-29</u>
学 院	<u>未来技术学院</u>	指导老师	<u>马 钊</u>

二〇二五年 一月



# 题目二十九 共享单车租赁时温度数据分析

## 1.1 题目：共享单车租赁时温度数据分析

### 【数据说明】

该数据集是 2011 年和 2012 年共享单车租借统计数据。

### 【字段说明】

- (1)instant:租借记录序号。
- (2)dteday:日期。
- (3)season:季节(1:springer,2:summer,3:fall,4: winter)。
- (4)yr:年(0:2011,1:2012)。
- (5)mnth:月(1~12)。
- (6)hr:时(0~23)。
- (7)holiday:是否是假期。
- (8)weekday:星期几。
- (9)workingday:是否工作日。
- (10)weathersit:天气。
  - 1 晴天(Clear)
  - 2 雾、云(Cloudy)
  - 3 小雨小雪(Light Rain)
  - 4 大雨(Heavy Rain)
- (11)temp:温度。
- (12)atemp:归一化温度。
- (13)hum:归一化湿度。
- (14)windspeed:归一化风速。
- (15)casual:临时用户。
- (16)registered:注册用户。

### 【任务】

- (1)用 pandas 库读取 bike\_day.csv 文件,查看前 5 行、后 2 行。
- (2)选择 instant、dteday、atemp、casual、registered 共 5 列数据,将缺失值全部丢弃处理,并将处理后结果导出到文本文件 bike\_atemp\_user.txt,要求数据之间用空格分隔,每行末尾包含换行符。
- (3)读取文本文件 bike\_atemp\_user.txt,计算列 casual 与列 registered 和(即 casual+registered),并作为一个新的列 cnt 添加到原始数据,导出到新的 Excel 文件 bike\_atempuser\_cnt.xlsx 中。
- (4)读取 Excel 文件 bike\_atemp\_user\_cnt.xlsx,统计列 atemp 的最大值 maxValue、最小值 minValue、平均值 meanValue。利用 category=[minValue,0.4,0.6,0.8, maxValue] 和 labels=['Cold','Cool','Warm','Hot']将 atemp 进行离散化;并将离散化结果作为一个新的列 Label

添加到原始数据集,并保存为 bike\_atemp\_user\_cnt\_result.csv。

(5)读取 bike\_atemp\_user\_cnt\_result.csv,按照列 Lable 分组,计算列 cnt 的均值,并以柱状图可视化显示该均值。要求以列 Lable 为 x 轴,轴刻度分别为 Cold、Cool、Warm、Hot,包括图例、图标题,填充颜色为蓝色,并保存为 bike\_atemp\_user\_cnt.png,要求分辨率不低于 300dpi。

## 1.2 需求分析

根据题目要求,任务主要涉及到数据读取、数据查询、数据分类汇总、数据计算及排序、数据可视化、数据导出等常规的数据分析操作步骤,可以调用 Pandas 的文件读写、数据分析等功能模块实现;各任务要求用函数形式实现,则需要设计各函数之间用参数传递实现各操作步骤之间的松耦合,进行模块化程序设计;各操作步骤需要用菜单实现功能选择,以及提供必要的输入输出等人机交互操作;此外,在程序中应提供必要的异常控制代码,保证程序的健壮性。

## 1.3 概要设计

根据需求分析,可以将该系统设计为“数据读取及预览”、“数据选择及导出”、“计算新列并导出”、“离散化数据并导出”、“数据可视化”五大功能模块,以及“功能选择”主菜单辅助模块。

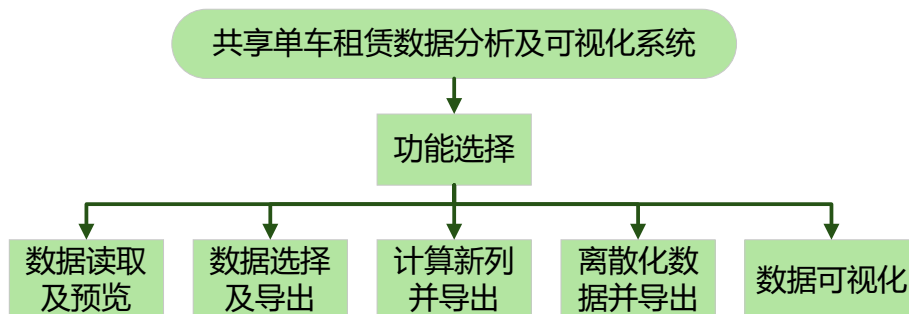


图 1 共享单车租赁数据分析及可视化系统概要设计

## 1.4 详细设计

### 1.4.1 主函数详细设计

#### 【分析】

在 Python 函数式编程中,主函数一般比较简洁,只提供函数调用。在本例中,主函数仅包含任务调用函数。

#### 【关键代码】

```
#####  
#主函数  
if __name__ == '__main__':  
    task()#调用功能选择函数
```

#####

#### 1.4.2. 功能选择模块详细设计

##### 【分析】

功能选择模块设计应比较简洁，只提供输入、功能处理和输出部分的函数调用。其中，各功能模块采用菜单式选择。另外，由于该实例中各任务之间存在先后顺序关系，后续任务的输入可能来源于前序任务的输出，因此在各步骤执行前还需要判断需要的数据源是否已经准备好，即前序任务是否已经执行完毕。

具体流程如下图所示：

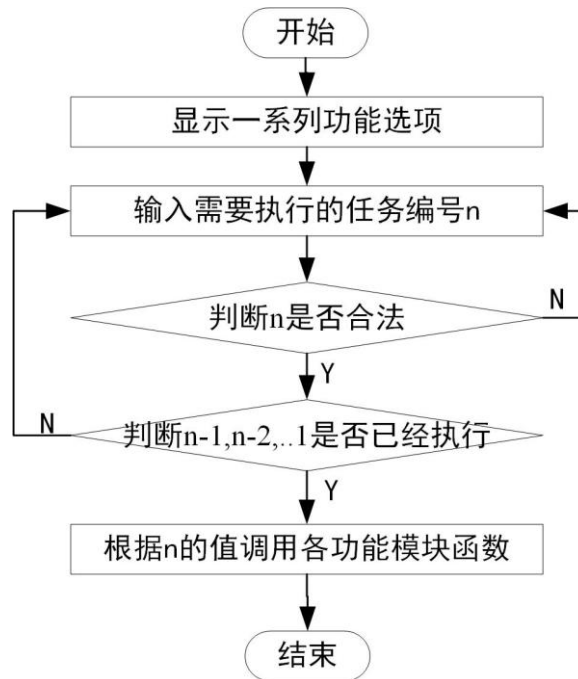


图 2 功能选择模块程序流程图

##### 【关键代码】

#####

# 系统主界面

def menu():

    print('【任务选择】\n')

#系统主界面

+----共享单车租赁数据分析及可视化系统-----+

|0、退出。

|1、数据读取及预览。

|2、数据选择及导出。

|3、计算新列并导出。

|4、离散化数据并导出。

|5、数据可视化。

+-----+

```
#####  
#功能选择模块  
def task():  
    while True:  
        menu() # 打印系统主界面  
        num = input("请输入任务选项: ")  
        if num == '1':  
            read_and_preview_data()  
        elif num == '2':  
            select_and_export_data()  
        elif num == '3':  
            calculate_and_export_data()  
        elif num == '4':  
            discretize_and_export_data()  
        elif num == '5':  
            visualize_data()  
        elif num == '0':  
            print('程序结束!')  
            break  
        else:  
            print('输入选项有误')  
            input("回车显示菜单")  
#####
```

#### 1.4.3 数据读取及预览模块详细设计

##### 【分析】

- (1) 用 pandas 库读取 bike\_day.csv 文件,查看前 5 行、后 2 行。
- (2) 该模块负责从文件或数据库中读取数据,并对数据进行初步预览。通常包括加载数据、显示前几行数据、查看数据的基本信息。

##### 【关键代码】

```
#####  
# 任务 1: 读取数据并查看前 5 行和后 2 行  
def read_and_preview_data():  
    file_name = input('请输入要打开的文件名 bike_day.csv:')  
    try:  
        df = pd.read_csv(file_name)  
        print("前 5 行数据: ")  
        print(df.head(5))  
        print("后 2 行数据: ")
```

```

        print(df.tail(2))
        print("任务 1 执行成功!")
    except Exception as e:
        print(f'任务 1 执行失败: {e}')
#####

```

#### 1.4.4 数据选择及导出模块详细设计

##### 【分析】

- (1) Pandas 提供了直接索引、基于标签索引、基于位置索引等数据选择功能。
- (2) 数据导出除了文件操作的 write(s)和 writelines(s)之外, 更为便捷的是 Pandas 提供的 to\_csv 和 to\_excel 两个功能。
- (3) 该模块选择 instant、dteday、atemp、casual、registered 共 5 列数据,将缺失值全部丢弃处理,并将处理后结果导出到文本文件 bike\_atemp\_user.txt,要求数据之间用空格分隔,每行末尾包含换行符。
- (4) 该模块允许用户选择特定的列或行, 并将选择的数据导出到新的文件中。用户可以根据需要筛选数据, 并保存为不同格式的文件 (如 CSV、Excel 等)

##### 【关键代码】

```

#####
# 任务 2: 选择特定列并处理缺失值
def select_and_export_data():
    file_name = input('请输入要打开的文件名 bike_day.csv:')
    try:
        df = pd.read_csv(file_name)
        df_selected = df[['instant', 'dteday', 'atemp', 'casual', 'registered']].dropna()
        df_selected.to_csv('bike_atemp_user.txt', sep=' ', index=False, lineterminator='\n')
        print("任务 2 执行成功!")
    except Exception as e:
        print(f'任务 2 执行失败: {e}')
#####

```

#### 1.4.5 计算新列并导出模块详细设计

##### 【分析】

- (1) 该模块用于在现有数据的基础上计算新的列。计算可以基于现有列的运算、转换或聚合。计算完成后, 用户可以将包含新列的数据导出。
- (2) 读取文本文件 bike\_atemp\_user.txt, 计算列 casual 与列 registered 和 (即 casual+registered), 并作为一个新的列 cnt 添加到原始数据, 导出到新的 Excel 文件 bike\_atempuser\_cnt.xlsx 中。

##### 【关键代码】

```
#####
# 任务 3: 计算新列并导出为 Excel 文件
def calculate_and_export_data():
    file_name = input('请输入要打开的文件名 bike_atemp_user.txt:')
    try:
        df = pd.read_csv(file_name, sep=' ')
        df['cnt'] = df['casual'] + df['registered']
        df.to_excel('bike_atemp_user_cnt.xlsx', index=False)
        print("任务 3 执行成功!")
    except Exception as e:
        print(f'任务 3 执行失败: {e}')
#####
```

#### 1. 4. 6 离散化数据并导模块详细设计

##### 【分析】

该模块读取 Excel 文件 bike\_atemp\_user\_cnt.xlsx,统计列 atemp 的最大值 maxValue、最小值 minValue、平均值 meanValue。利用 category=[minValue,0.4,0.6,0.8,maxValue] 和 labels=['Cold','Cool','Warm','Hot']将 atemp 进行离散化;并将离散化结果作为一个新的列 Label 添加到原始数据集,并保存为 bike\_atemp\_user\_cnt\_result.csv。

##### 【关键代码】

```
#####
# 任务 4: 离散化数据并导出为 CSV 文件
def discretize_and_export_data():
    file_name = input('请输入要打开的文件名 bike_atemp_user_cnt.xlsx:')
    try:
        df = pd.read_excel(file_name)
        max_value = df['atemp'].max()
        min_value = df['atemp'].min()
        mean_value = df['atemp'].mean()
        category = [min_value, 0.4, 0.6, 0.8, max_value]
        labels = ['Cold', 'Cool', 'Warm', 'Hot']
        df['Label'] = pd.cut(df['atemp'], bins=category, labels=labels, right=False)
        df.to_csv('bike_atemp_user_cnt_result.csv', index=False)
        print("任务 4 执行成功!")
    except Exception as e:
        print(f'任务 4 执行失败: {e}')
#####
```

#### 1. 4. 7 数据可视化模块详细设计

##### 【分析】



数据可视化,是数据分析的重要组成部分。在 Python 生态系统中,包括诸如 matplotlib、Seaborn、HoloViews、Altair、PyQtGraph、ggplot、Bokeh、pygal、VisPy、NetworkX、Plotly、geoplotlib、folium、Gleam、vincent、mpld3、python-igraph、missingno、Mayavi2、Leather 等等众多的扩展库实现数据可视化功能。

该模块读取 bike\_atemp\_user\_cnt\_result.csv,按照列 Lable 分组,计算列 cnt 的均值,并以柱状图可视化显示该均值。要求以列 Lable 为 x 轴,轴刻度分别为 Cold、Cool、Warm、Hot,包括图例、图标题,填充颜色为蓝色,并保存为 bike\_atemp\_user\_cnt.png

### 【关键代码】

```
#####  
# 任务 5: 可视化数据并保存为图片  
def visualize_data():  
    file_name = input('请输入要打开的文件名 bike_atemp_user_cnt_result.csv:')  
    try:  
        df = pd.read_csv(file_name)  
        df_grouped = df.groupby('Label')['cnt'].mean()  
  
        plt.figure(figsize=(10, 6))  
        df_grouped.plot(kind='bar', color='blue')  
        plt.xlabel('Label', fontsize=12)  
        plt.ylabel('Average Count', fontsize=12)  
        plt.title('Average Bike Rentals by Temperature Category', fontsize=16)  
        plt.xticks(rotation=0)  
        plt.savefig('bike_atemp_user_cnt.png', dpi=300)  
        plt.show()  
  
        print("任务 5 执行成功!")  
    except Exception as e:  
        print(f'任务 5 执行失败: {e}')  
#####
```

## 9.4 完整参考代码

```
#####  
Created on 30/12/2024  
@author: Kanghui Zeng  
运行环境: Windows 11 系统, Python 3.9  
import pandas as pd  
import matplotlib.pyplot as plt  
import os  
# 防止中文出现乱码
```

---

```
plt.rcParams['font.sans-serif'] = ['SimHei'] # 黑体
plt.rcParams['axes.unicode_minus'] = False
# 任务 1: 读取数据并查看前 5 行和后 2 行
def read_and_preview_data():
    file_name = input('请输入要打开的文件名 bike_day.csv:')
    try:
        df = pd.read_csv(file_name)
        print("前 5 行数据: ")
        print(df.head(5))
        print("后 2 行数据: ")
        print(df.tail(2))
        print("任务 1 执行成功!")
    except Exception as e:
        print(f'任务 1 执行失败: {e}')
# 任务 2: 选择特定列并处理缺失值
def select_and_export_data():
    file_name = input('请输入要打开的文件名 bike_day.csv:')
    try:
        df = pd.read_csv(file_name)
        df_selected = df[['instant', 'dteday', 'atemp', 'casual', 'registered']].dropna()
        df_selected.to_csv('bike_atemp_user.txt', sep=' ', index=False, lineterminator='\n')
        print("任务 2 执行成功!")
    except Exception as e:
        print(f'任务 2 执行失败: {e}')
# 任务 3: 计算新列并导出为 Excel 文件
def calculate_and_export_data():
    file_name = input('请输入要打开的文件名 bike_atemp_user.txt:')
    try:
        df = pd.read_csv(file_name, sep=' ')
        df['cnt'] = df['casual'] + df['registered']
        df.to_excel('bike_atemp_user_cnt.xlsx', index=False)
        print("任务 3 执行成功!")
    except Exception as e:
        print(f'任务 3 执行失败: {e}')
# 任务 4: 离散化数据并导出为 CSV 文件
def discretize_and_export_data():
    file_name = input('请输入要打开的文件名 bike_atemp_user_cnt.xlsx:')
    try:
        df = pd.read_excel(file_name)
        max_value = df['atemp'].max()
```

```
minValue = df['atemp'].min()
meanValue = df['atemp'].mean()
category = [minValue, 0.4, 0.6, 0.8, maxValue]
labels = ['Cold', 'Cool', 'Warm', 'Hot']
df['Label'] = pd.cut(df['atemp'], bins=category, labels=labels, right=False)
df.to_csv('bike_atemp_user_cnt_result.csv', index=False)
print("任务 4 执行成功!")
except Exception as e:
    print(f'任务 4 执行失败: {e}')
# 任务 5: 可视化数据并保存为图片
def visualize_data():
    file_name = input('请输入要打开的文件名 bike_atemp_user_cnt_result.csv:')
    try:
        df = pd.read_csv(file_name)
        df_grouped = df.groupby('Label')['cnt'].mean()

        plt.figure(figsize=(10, 6))
        df_grouped.plot(kind='bar', color='blue')
        plt.xlabel('Label', fontsize=12)
        plt.ylabel('Average Count', fontsize=12)
        plt.title('Average Bike Rentals by Temperature Category', fontsize=16)
        plt.xticks(rotation=0)
        plt.savefig('bike_atemp_user_cnt.png', dpi=300)
        plt.show()
        print("任务 5 执行成功!")
    except Exception as e:
        print(f'任务 5 执行失败: {e}')
# 系统主界面
def menu():
    print('【任务选择】\n'
          '----共享单车租赁数据分析及可视化系统-----+\n'
          '0、退出。 \n'
          '1、数据读取及预览。 \n'
          '2、数据选择及导出。 \n'
          '3、计算新列并导出。 \n'
          '4、离散化数据并导出。 \n'
          '5、数据可视化。 \n'
          '+-----+')
# 功能选择模块
def task():
```

```
while True:
    menu() # 打印系统主界面
    num = input("请输入任务选项: ")
    if num == '1':
        read_and_preview_data()
    elif num == '2':
        select_and_export_data()
    elif num == '3':
        calculate_and_export_data()
    elif num == '4':
        discretize_and_export_data()
    elif num == '5':
        visualize_data()
    elif num == '0':
        print('程序结束!')
        break
    else:
        print('输入选项有误')
        input("回车显示菜单")
# 主函数
if __name__ == '__main__':
    task()
```

---

---

## 9.5 程序运行截图

```
E:\soft\Python\python.exe E:\code\python\Design\No.29\bike.py
【任务选择】
+---共享单车租赁数据分析及可视化系统-----+
|0、退出。
|1、数据读取及预览。
|2、数据选择及导出。
|3、计算新列并导出。
|4、离散化数据并导出。
|5、数据可视化。
+-----+
请输入任务选项: 2
请输入要打开的文件名bike_day.csv:bike_day.csv
任务2执行成功!
回车显示菜单
【任务选择】
+---共享单车租赁数据分析及可视化系统-----+
|0、退出。
|1、数据读取及预览。
|2、数据选择及导出。
|3、计算新列并导出。
|4、离散化数据并导出。
|5、数据可视化。
+-----+
请输入任务选项: 3
请输入要打开的文件名bike_atemp_user.txt:bike_atemp_user.txt
任务3执行成功!
回车显示菜单
【任务选择】
+---共享单车租赁数据分析及可视化系统-----+
|0、退出。
|1、数据读取及预览。
|2、数据选择及导出。
|3、计算新列并导出。
|4、离散化数据并导出。
|5、数据可视化。
+-----+
请输入任务选项: 4
请输入要打开的文件名bike_atemp_user_cnt.xlsx:bike_atemp_user_cnt.xlsx
任务4执行成功!
回车显示菜单
【任务选择】
+---共享单车租赁数据分析及可视化系统-----+
|0、退出。
|1、数据读取及预览。
|2、数据选择及导出。
|3、计算新列并导出。
|4、离散化数据并导出。
|5、数据可视化。
+-----+
请输入任务选项: 5
请输入要打开的文件名bike_atemp_user_cnt_result.csv:bike_atemp_user_cnt_
任务5执行成功!
回车显示菜单
【任务选择】
+---共享单车租赁数据分析及可视化系统-----+
|0、退出。
|1、数据读取及预览。
|2、数据选择及导出。
|3、计算新列并导出。
|4、离散化数据并导出。
|5、数据可视化。
+-----+
请输入任务选项: 0
程序结束!

Process finished with exit code 0
```

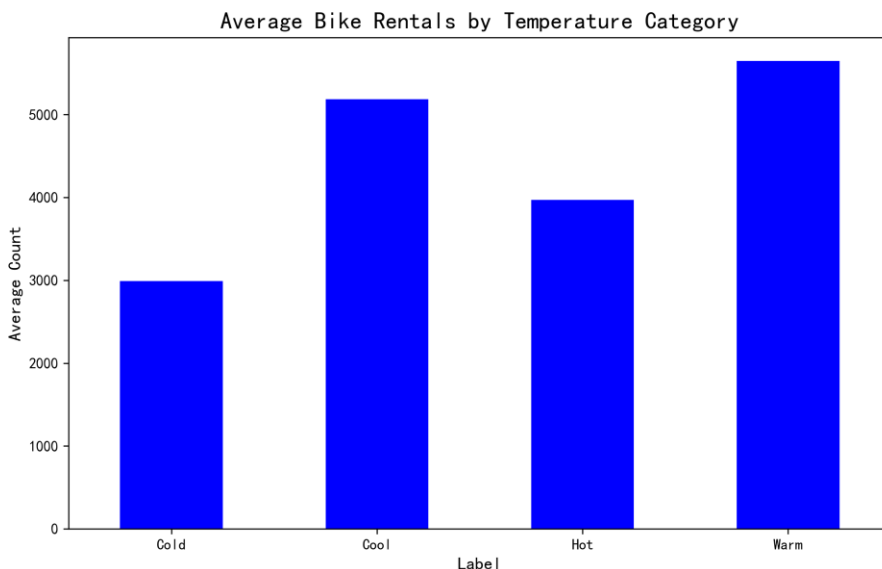


图 3 不同天气下的共享单车日均租赁人数

## 9.6 结果分析

通过对共享单车租赁数据的分析，我们可以得出以下结论：

1. 数据预览：通过读取并预览数据，我们可以初步了解数据的结构和内容。前 5 行和后 2 行的数据展示了数据的基本情况，包括日期、温度、用户类型等信息。
2. 数据选择与导出：选择了 instant、dteday、atemp、casual、registered 五列数据，并处理了缺失值。将处理后的数据导出为文本文件，便于后续分析。
3. 计算新列并导出：计算了 casual 和 registered 的和，作为新的列 cnt，并将结果导出为 Excel 文件。这一步为后续的离散化和可视化提供了基础数据。
4. 数据离散化：根据 atemp 的最大值、最小值和平均值，将温度数据离散化为 Cold、Cool、Warm、Hot 四个类别，并将离散化结果添加到数据集中。离散化后的数据更易于分类和分析。
5. 数据可视化：通过柱状图展示了不同温度类别下的平均单车租赁数量。结果显示，**温度对单车租赁数量有显著影响，适宜的天气（Warm 和 Cool）下租赁数量较高，极端的天气（Cold 和 Hot）下租赁数量较低。**

通过以上分析，我们可以更好地理解温度对共享单车租赁的影响，为共享单车的运营和管理提供数据支持。

# 课程设计总结

## 问题：

通过这几天的 Python 语言课程设计，让我对 Python 语言的知识进行了一次系统的回顾。通过这次上机实习，让我综合应用起整个学期的基础知识，编写程序渐渐地建立起了规模，从做题练习走向实际应用的层面。感受到了编程在实际生活中起到的重要作用。

## 解决方法：

这次的实习对我们来说很有挑战，但是这也能锻炼我们的能力，这是一个提升我编程能力的好机会。在实习过程中我遇到了各种各样的问题，常常事与愿违。对于解决方法显而易见的问题，能自己解决的尽量自己解决，实在感到自己余力不足，就与同学们一起探讨。

## 不足：

在这次课程设计中，我发现自己在以下几个方面还存在不足：

1. 对于一些高级的 Python 特性和库的使用还不够熟练。
2. 在编写代码时，有时会忽略代码的可读性和规范性。
3. 对于一些复杂问题的解决思路还不够清晰，容易陷入细节而忽略整体。

## 展望：

通过这次课程设计，我深刻认识到编程能力的重要性和自身的不足。在未来的学习和实践中，我将继续努力，提升自己的编程水平：

1. 加强对 Python 高级特性和库的学习，提升编程效率。
2. 注重代码的可读性和规范性，养成良好的编程习惯。
3. 多参与实际项目，锻炼解决复杂问题的能力，提升综合素质。

总之，这次 Python 课程设计不仅让我巩固了所学知识，还让我认识到自身的不足和努力的方向。未来，我将继续努力，不断提升自己的编程能力，为成为一名优秀的程序员而奋斗。