

实验报告三

22010022065肖予纯

2024 年 9 月 9 日

1 Github

<https://github.com/xiaoyuchun666/homework-tools.git>

2 命令行环境

2.1 练习内容

1. 我们可以使用类似 `ps aux | grep` 这样的命令来获取任务的 `pid`，然后您可以基于 `pid` 来结束这些进程。但我们其实有更好的方法来做这件事。在终端中执行 `sleep 10000` 这个任务。然后用 `Ctrl-Z` 将其切换到后台并使用 `bg` 来继续允许它。现在，使用 `pgrep` 来查找 `pid` 并使用 `pkill` 结束进程而不需要手动输入 `pid`。(提示：使用 `-af` 标记)。
2. 请完成这个 `tmux` 教程参考这些步骤来学习如何自定义 `tmux`
3. 创建一个 `dc` 别名，它的功能是当我们错误的将 `cd` 输入为 `dc` 时也能正确执行。
4. 执行 `history | awk '1 == "" {print substr(0,2)}' | sort | uniq -c | sort -n | tail -n 10` 来获取您最常用的十条命令，尝试为它们创建别名。
注意：这个命令只在 `Bash` 中生效，如果您使用 `ZSH`，使用 `history 1` 替换 `history`。

2.2 代码部分

1.

```
sleep 10000 &
```

```
Ctrl-Z
```

```
bg
```

```
pgrep -af sleep
```

```
pkill -f sleep
```

2.

```
tmux new -s my_session
```

```
Ctrl+b d
```

```
tmux attach-session -t my_session
```

```
tmux split-window
```

```
tmux split-window -h
```

```
tmux select-pane -U
```

```
tmux select-pane -D
```

```
tmux select-pane -L
```

```
tmux select-pane -R
```

```
tmux swap-pane -U
```

```
tmux swap-pane -D
```

```
tmux new-window -n <window-name>
```

```
tmux select-window -t <window-number>
```

```
tmux select-window -t <window-name>
```

```
tmux rename-window <new-name>
```

```
Ctrl+b c/p/n/<number>/w/ ,
```

```
tmux list-keys
```

```
tmux list-commands
```

```
tmux info
```

```
tmux source-file ~/.tmux.conf
```

```
3.  
alias dc=cd  
alias ll='ls -l'
```

2.3 结果

图1: 1and2、图2: 3、图3: 4

3 Python入门基础

3.1 练习内容

1. hello world
2. print()
3. if条件判断
4. while循环计算100以内偶数和
5. for循环
6. 函数应用斐波那契数列
7. 水仙花数
8. lambda
9. 列表、元组、字典
10. 文件读写

3.2 代码部分

```
1.  
print('hello world!')
```

```
if 3 > 0:
    print('OK')
    print('yes')
```

```
x = 3
y = 4
print(x + y)
```

```
2.
print('hello ', 'world!')
print('hello ' + 'world!')
print('hello ', 'world ', sep='***')
print('#' * 50)
print('how are you?', end='')
```

```
3.
import random

num = random.randint(1, 10)
answer = int(input('guess a number: '))
if answer > num:
    print('answer is bigger')
elif answer < num:
    print('answer is smaller')
else:
    print('right!')

print('the number:', num)
```

```
4.
sum100 = 0
counter = 0
```

```

while counter < 100:
    counter += 1

    if counter % 2 == 1:
        continue
    sum100 += counter

print(sum100)

```

5.

```

astr = 'hello '
for ch in astr:
    print(ch)
for i in range(10):
    print(i)

```

6.

```

def gen_fib(l):
    fib = [0, 1]

    for i in range(l - len(fib)):
        fib.append(fib[-1] + fib[-2])

    return fib

```

```

a = gen_fib(10)
print(a)
print('-' * 50)
n = int(input("length: "))
print(gen_fib(n))

```

7.

```

for i in range(100,1000):

```

```

s=str(i)
one=int(s[-1])
ten=int(s[-2])
hun=int(s[-3])
if i == one**3+ten**3+hun**3:
print(i)

```

8.

```

Max=lambda x,y:x*(x>=y)+y*(y>x)
Min=lambda x,y:x*(x<=y)+y*(y<x)

```

```

a=int(input('1:'))
b=int(input('2:'))

```

```

print(Max(a,b))
print(Min(a,b))

```

9.

```

**alist = [10, 20, 30, 'bob', 'alice', [1,2,3]]
len(alist)
alist[-1] # [1,2,3]
alist[-1][-1] # 3
[1,2,3][-1] # 3
alist[-2][2] # i
alist[3:5] # ['bob', 'alice']
10 in alist # True
'o' in alist # False
100 not in alist # True
alist[-1] = 100
alist.append(200)

```

```

classmates = ['Michael', 'Bob', 'Tracy']
print(len(classmates))

```

```
scores = [95, 75, 85]
cc=dict(zip(classmates,scores))
print(cc['Michael'])
```

10.

```
f = open('/tmp/passwd')
data = f.read()
print(data)
data = f.read()
f.close()
```

```
f.tell()
f.readline()
f.tell()
f.seek(0, 0)
```

```
f.tell()
f.close()
```

4 Python视觉应用

4.1 练习内容

1. 练习1

以“populations.npz”为素材，完成题目的要求。（要求在绘图之前，先进行数据处理：删去全为“nan”的行，然后按年份“从小到大”排序）

绘制总人口线形图。x轴是“年份”，从起始年开始，每隔4年绘制一个刻度。

2. 练习2

创建一个有50个元素的Series对象，其values数组中的数据随机生成，数据总体上满足均值为1000，标准差为200的正态分布。

(1) 自定义异常值为：小于 $QL-1.25IQR$ 或大于 $QU+1.25IQR$ 的值。绘制箱线图，检测生成的数据中是否包含大于上限和小于下限的异常值，并且要求这两类异常值都要有。如果不满足要求，那么就重新生成数据，直到满足要求为止；

(2) 利用箱线图获取并输出异常值的索引；

(3) 编写一个通用函数，其功能为将一个Series对象中大于上限的异常值用QU替换，而小于下限的异常值用QL替换。（原地操作）

(4) 用该函数处理满足(1)要求的Series对象，输出QU、QL的值。然后，创建一个如下所示的DataFrame对象，其index为异常值的索引，Before列上的数据为替换前的值，After列上的数据为替换后的值。最后，输出该DataFrame对象。

3. 练习3

以“iris.npz”为素材，绘制散点图，包括4个子图（ 2×2 ）。4个子图分别为“萼片(sepal)长度-花瓣(petal)长度”、“萼片宽度-花瓣宽度”、“萼片长度-萼片宽度”、“花瓣长度-花瓣宽度”散点图。（为了便于分析数据，在每个子图中，要求用3种不同的颜色表示3种不同类别的花）。

4.2 代码部分

4.3 结果


```
xiaoyuchun@hadoop01:~/Desktop
File Edit View Search Terminal Help
500      2349 0.0 1.9 324384 19920 ?      S   01:45  0:00 python /usr/sh
500      2353 0.0 0.9 260904  9684 ?      S   01:45  0:00 bluetooth-appl
root     2360 0.0 0.5  80052  5388 ?      S   01:45  0:00 /usr/sbin/pack
500      2361 0.0 0.3 264028  3116 ?      Ss  01:45  0:00 gnome-screensa
500      2375 0.0 0.2 232488  2636 ?      Sl  01:45  0:00 /usr/libexec/g
500      2379 0.0 0.2 150936  2612 ?      S   01:45  0:00 /usr/libexec/g
500      2389 0.0 2.1 400652 21908 ?      S   01:45  0:00 /usr/bin/gnote
500      2390 0.0 1.3 392416 13308 ?      S   01:45  0:00 /usr/libexec/g
500      2391 0.0 1.5 461256 15916 ?      S   01:45  0:00 /usr/libexec/c
500      2392 0.0 1.0 291080 10348 ?      S   01:45  0:00 /usr/libexec/n
500      2414 0.0 0.2 137296  2488 ?      S   01:45  0:00 /usr/libexec/g
500      2446 0.0 1.3 295504 13864 ?      Sl  01:45  0:00 /usr/bin/gnome
500      2447 0.0 0.0   8224   708 ?      S   01:45  0:00 gnome-pty-help
500      2448 0.0 0.1 108304  1908 pts/0    Ss  01:45  0:00 /bin/bash
500      2477 0.0 0.1 110232  1172 pts/0    R+  01:51  0:00 ps aux
[xiaoyuchun@hadoop01 Desktop]$ sleep 10000 &
[1] 2478
[xiaoyuchun@hadoop01 Desktop]$ bg
bash: bg: job 1 already in background
[xiaoyuchun@hadoop01 Desktop]$ pgrep -f sleep
2478
[xiaoyuchun@hadoop01 Desktop]$ kill -f sleep
[1]+  Terminated                  sleep 10000
[xiaoyuchun@hadoop01 Desktop]$
```

```
xiaoyuchun@LAPTOP-8H9PA221:~$
189: setaf: (string) \033[%?%p1%{0}%<%t3%p1%d%e%p1%{16}%<%t9%p1%{0}%~%d%e38;5;%p1%d%;m
190: setrgbb: [missing]
191: setrgbf: [missing]
192: sgr0: (string) \033(B\033[m
193: sitm: (string) \033[3m
194: smacs: (string) \033(0
195: smcup: (string) \033[210H9h\033[22;0;0t
196: smxx: (string) \033[7ih\033=
197: smso: (string) \033[7m
198: smul: (string) \033[4m
199: smxx: (string) \033[9m
200: Ss: (string) \033[%p1%d q
201: Tc: [missing]
202: tsl: (string) \033]0;
203: U8: [missing]
204: vpa: (string) \033[%i%p1%dd
205: xenl: (flag) true
206: XT: (flag) true
xiaoyuchun@LAPTOP-8H9PA221:~$
xiaoyuchun@LAPTOP-8H9PA221:~$ tmux select-pane -D
xiaoyuchun@LAPTOP-8H9PA221:~$
xiaoyuchun@LAPTOP-8H9PA221:~$ tmux select-pane -U
xiaoyuchun@LAPTOP-8H9PA221:~$ tmux select-pane -L
xiaoyuchun@LAPTOP-8H9PA221:~$
```

```
ipy_session0: bash*
*LAPTOP-8H9PA221* 21:29 08-Sep-20
```

图 1: 1

```

xiaoyuchun@LAPTOP-8H9PA221:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0
sed -e '\''s/^\s*[0-9]\+\s*//;s/[:&]\s*alert$//'\''
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ls='ls --color=auto'
xiaoyuchun@LAPTOP-8H9PA221:~$ alias dc=cd
xiaoyuchun@LAPTOP-8H9PA221:~$ alias ll='ls -l'
xiaoyuchun@LAPTOP-8H9PA221:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0
sed -e '\''s/^\s*[0-9]\+\s*//;s/[:&]\s*alert$//'\''
alias dc='cd'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -l'
alias ls='ls --color=auto'

```

图 2: 3

```

xiaoyuchun@LAPTOP-8H9PA221:~$ history | awk '{ $1="" ; print substr($0,2) }' | sort |
10
1 unalias ll
2 alias dc=cd
2 tmux kill-session -t 0
2 tmux kill-session -t 1
2 tmux new -s l
2 tmux new -s my_session
4 tmux attach-session -t my_session
5 alias
8 tmux
9 tmux ls
xiaoyuchun@LAPTOP-8H9PA221:~$ alias tmux=tt
xiaoyuchun@LAPTOP-8H9PA221:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo e
sed -e '\''s/^\s*[0-9]\+\s*//;s/[:&]\s*alert$//'\''
alias dc='cd'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -l'
alias ls='ls --color=auto'
alias tmux='tt'
xiaoyuchun@LAPTOP-8H9PA221:~$ unalias tt
-bash: unalias: tt: not found
xiaoyuchun@LAPTOP-8H9PA221:~$ unalias tmux

```

图 3: 4

```

1 import matplotlib.pyplot as plt
2 import matplotlib
3 import numpy as np
4 import pandas as pd
5 matplotlib.rcParams['font.family']='FangSong'
6 datas=np.load( file: "E:\学习资料\三春\python科学计算\实验二\data\populations.npz",allow_pickle=True)
7 data=datas['data']
8 y=datas['feature_names']
9 df=pd.DataFrame(data=data,index=None,columns=y,dtype=None)
10 df.dropna(axis=0,how='all',inplace=True)
11 df.sort_values(by=['时间'],axis=0,ascending=True,inplace=True)
12 plt.figure(figsize=(8,4))
13 a=df['时间']
14 b=df['年末总人口(万人)']
15 c=list()
16 for i in range(len(a)):
17     c.append(float(a[i].rstrip("年")))
18 c.sort()
19 plt.plot('args: c,b,'ko--')
20 plt.xlabel("年份")
21 plt.ylabel("总人口")
22 xlable = list(range(1996, 2016, 4))
23 xl = [str(i) + "年" for i in xlable]
24 plt.xticks(xlable, xl)
25 plt.title("总人口线形图")
26 plt.legend(["总人口"])
27 plt.show()

```

图 4: 练习1代码

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 def outrange(ser):#进行替换
5     QU=ser.quantile(upper)
6     QL=ser.quantile(lower)
7     IQR=QU-QL
8     ser[ser > QU + whis*IQR]=QU
9     ser[ser < QL - whis*IQR]=QL
10    return ser
11 def yes(outlier,QU,QL,whis,IQR):#判断超过上界和下届的值都存在
12     for item in outlier:
13         for a in outlier:
14             if item > QU+whis*IQR:
15                 if a < QL-whis*IQR:
16                     return True
17     return False
18 def cc():
19     data = np.round(np.random.normal(loc=1000, scale=200, size=50), decimals= 2)
20     ser1 = pd.Series(data)
21     QU = ser1.quantile(upper)
22     QL = ser1.quantile(lower)
23     IQR = QU - QL
24     p = plt.boxplot(x=ser1, notch=True, whis=1.25, labels=['DATA'])
25     outlier1 = p['fliers'][0].get_ydata()
26     if yes(outlier1,QU,QL,whis,IQR):
27         return False
28     else:
29         return True
30 upper=0.75
31 lower=0.25
32 whis=1.25
33 plt.figure(figsize=(5,4))
34 data = np.round(np.random.normal(loc=1000, scale=200, size=50), decimals= 2)
35 ser1 = pd.Series(data)
upper=0.75
lower=0.25
whis=1.25
plt.figure(figsize=(5,4))
data = np.round(np.random.normal(loc=1000, scale=200, size=50), decimals= 2)
ser1 = pd.Series(data)
QU = ser1.quantile(upper)
QL = ser1.quantile(lower)
IQR = QU - QL
p = plt.boxplot(x=ser1, notch=True, whis=1.25, labels=['DATA'])
plt.cla()
outlier1 = p['fliers'][0].get_ydata()
#循环，直到找到符合 (1) 要求的数据
while cc():
    plt.cla()
plt.show()#显示图
#得出index
index= ser1[ser1.isin(outlier1)].index
print('原对象: \n',ser1)
col1=ser1[index]#原始异常值
print('上四分位数QU',QU,'下四分位数QL',QL,'四分位间距IQR: ',IQR)
print('异常值: ',outlier1)
print('异常值索引: ',index)
ser1=outrange(ser1)#进行替换
print('替换后的对象: \n',ser1)
col2=ser1[index]#替换后的值
df=pd.DataFrame( data= {'Before':col1,'After':col2},index=index)
print(df)

```

图 5: 练习2代码

图 6: 练习3代码(github上)

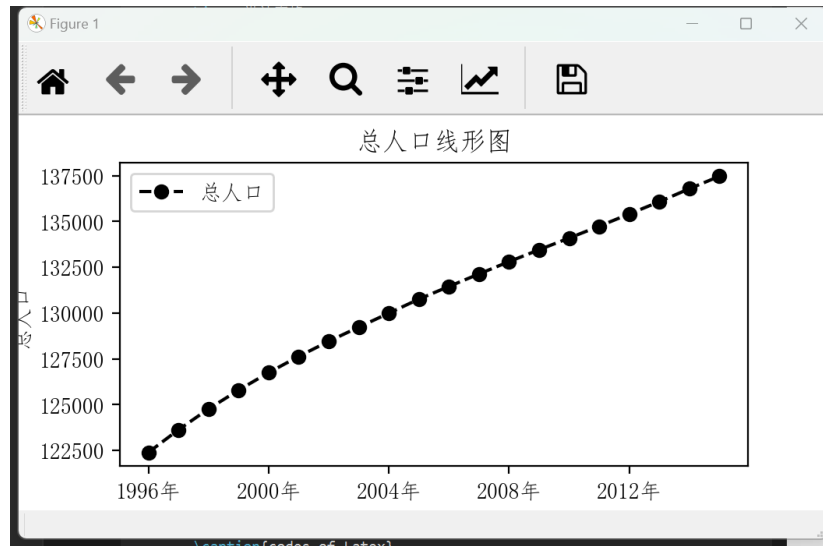


图 7: 练习1结果

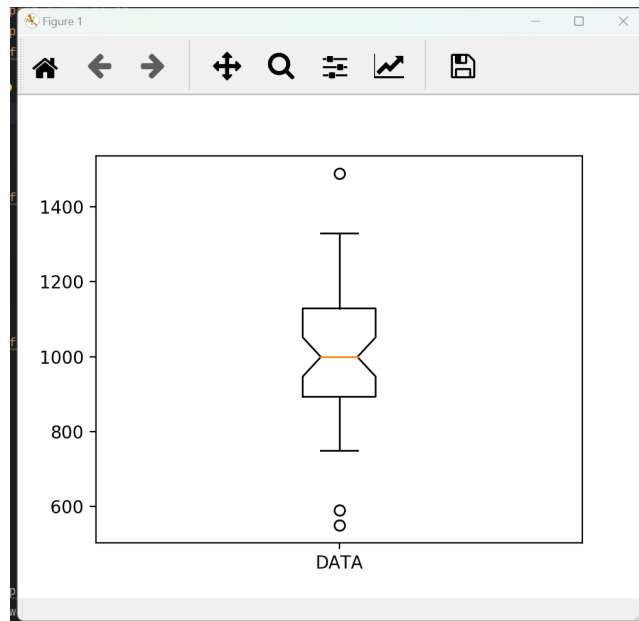


图 8: 练习2结果

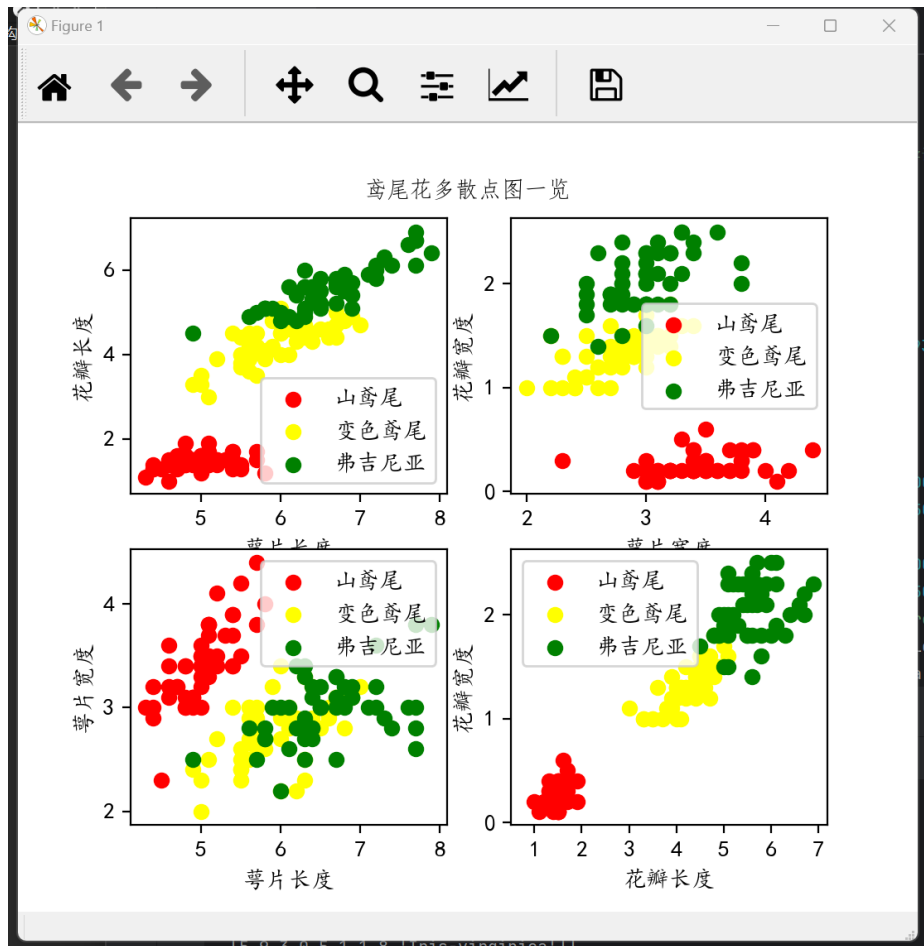


图 9: 练习3结果