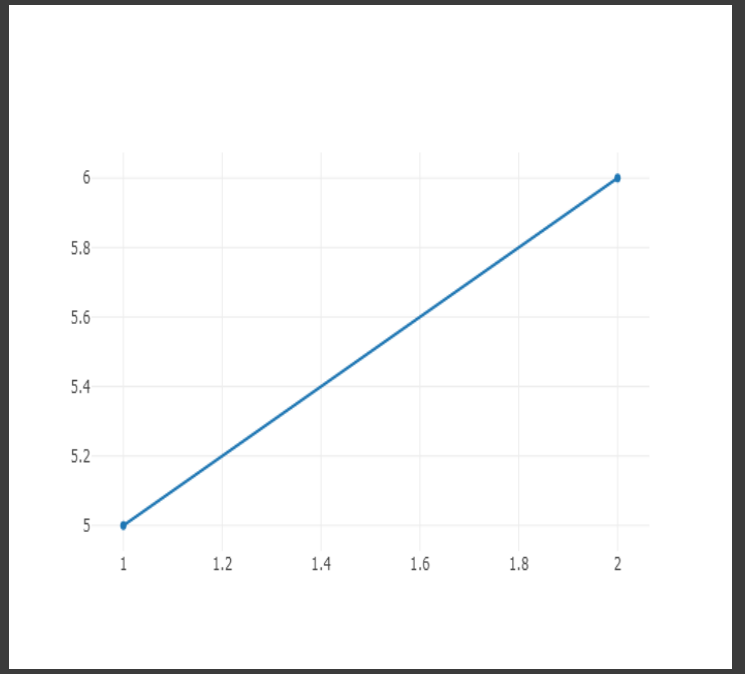
7.3 数据可视化—plotly模块(4.0版)

- from plotly.offline import plot
- import plotly.express as px
- import pandas as pd

- dataset=pd.DataFrame({"x":[
  1,2],"y":[5,6]})
- figure = px.line(dataset,x="x",y="y")
- plot(figure)

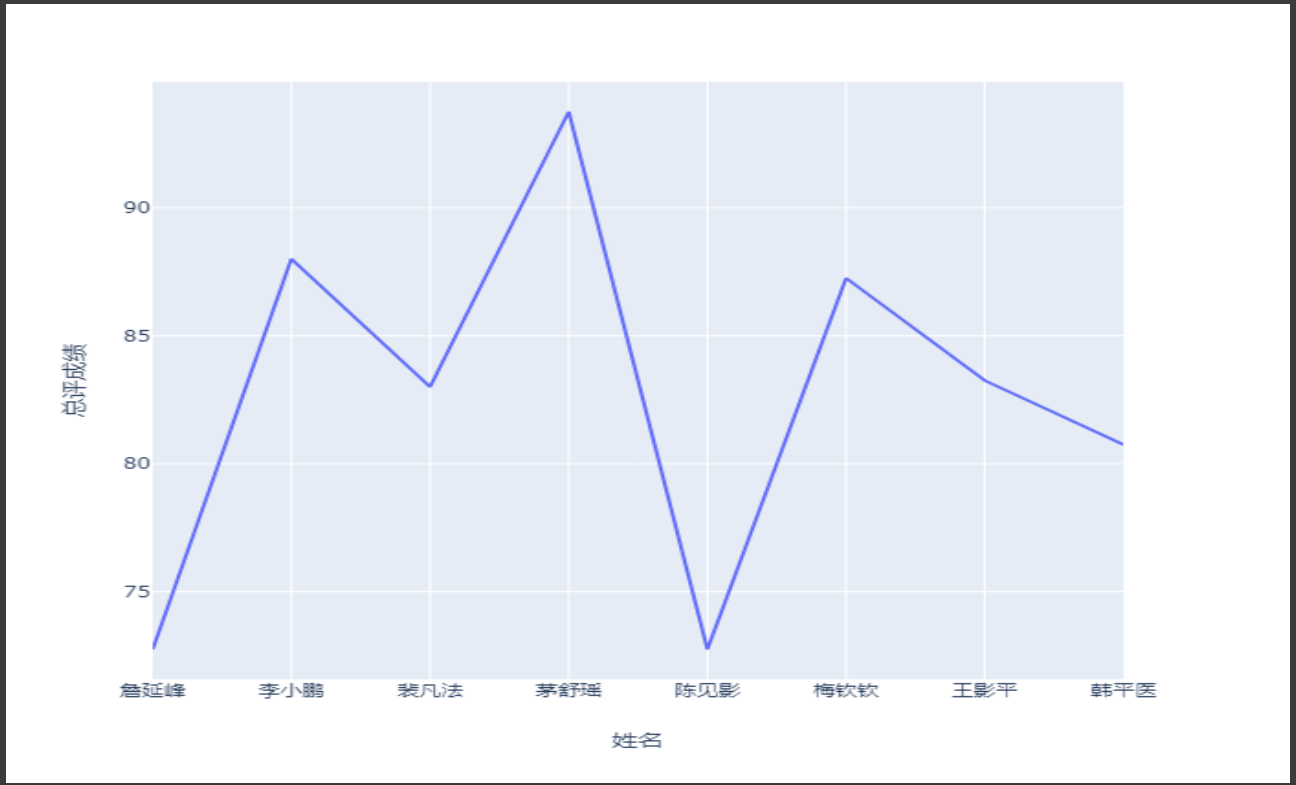


折线图

- from plotly.offline import plot
- import plotly.express as px
- import pandas as pd

- data = pd.read_csv("score.csv",encoding="GBK")
- data["总评成绩 "]=data["笔试"]*0.5+data["平时"]*0.25+data["实验"]*0.25
- figure = px.line(data,x="姓名",y="总评成绩 ")
- plot(figure)

输出图形



同时绘制笔试和总评分的折线图

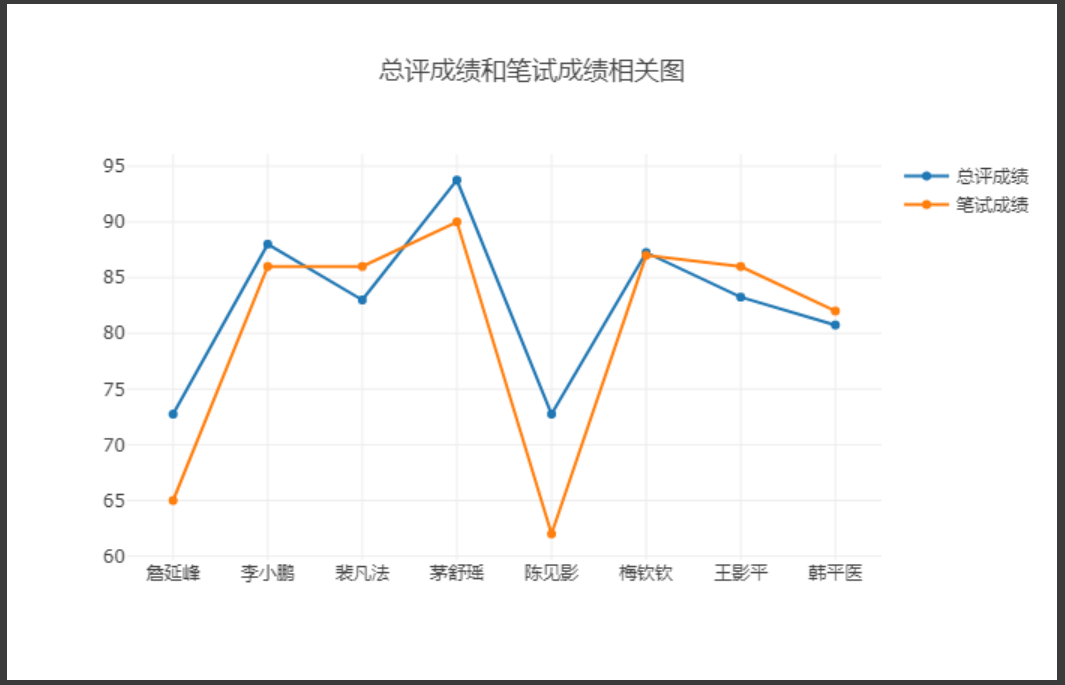
- from plotly.offline import plot
- import plotly.graph_objs as go

- import pandas as pd
- data = pd.read_csv("score.csv",encoding="GBK")
- data["总评成绩 "]=data["笔试"]*0.5+data["平时"]*0.25+data["实验"]*0.25
- xdata=data["姓名"].tolist() #取姓名这 一列，变列表
- ydata1=data["总评成绩 "].tolist() #取总评成绩这一列，变列表
- ydata2=data["笔试"].tolist() #取笔试成绩这一列，变列表

- trace0=go.Scatter(x=xdata,y=ydata1,name="总评成绩") #总评折线
- trace1=go.Scatter(x=xdata,y=ydata2,name="笔试成绩") #笔试折线
- mylayout=go.Layout(title="总评成绩和笔试成绩相关图") #图的标题

- fig=go.Figure(data=[trace0,trace1],layout=mylayout)
- plot(fig)

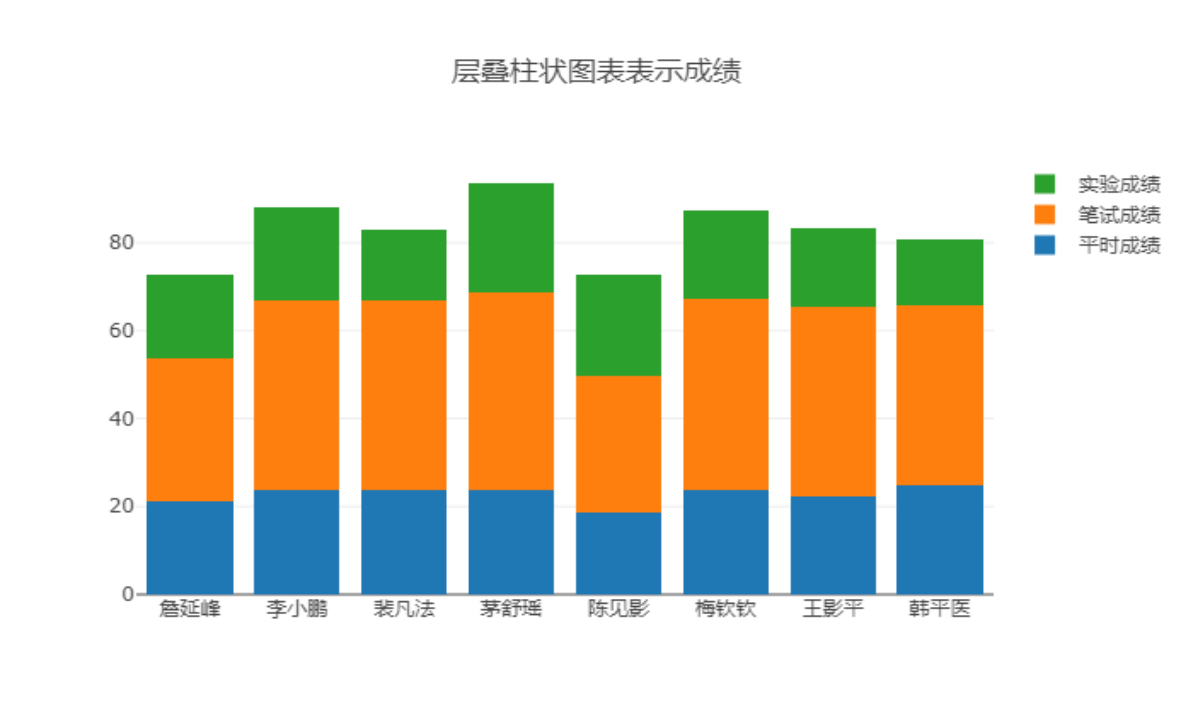
显示图形


总评成绩和笔试成绩相关图
95
90
85
80
75
70
65
60
詹延峰
李小鹏
裴凡法
茅舒瑶
陈见影
梅钦钦
王影平
韩平医
总评成绩
笔试成绩

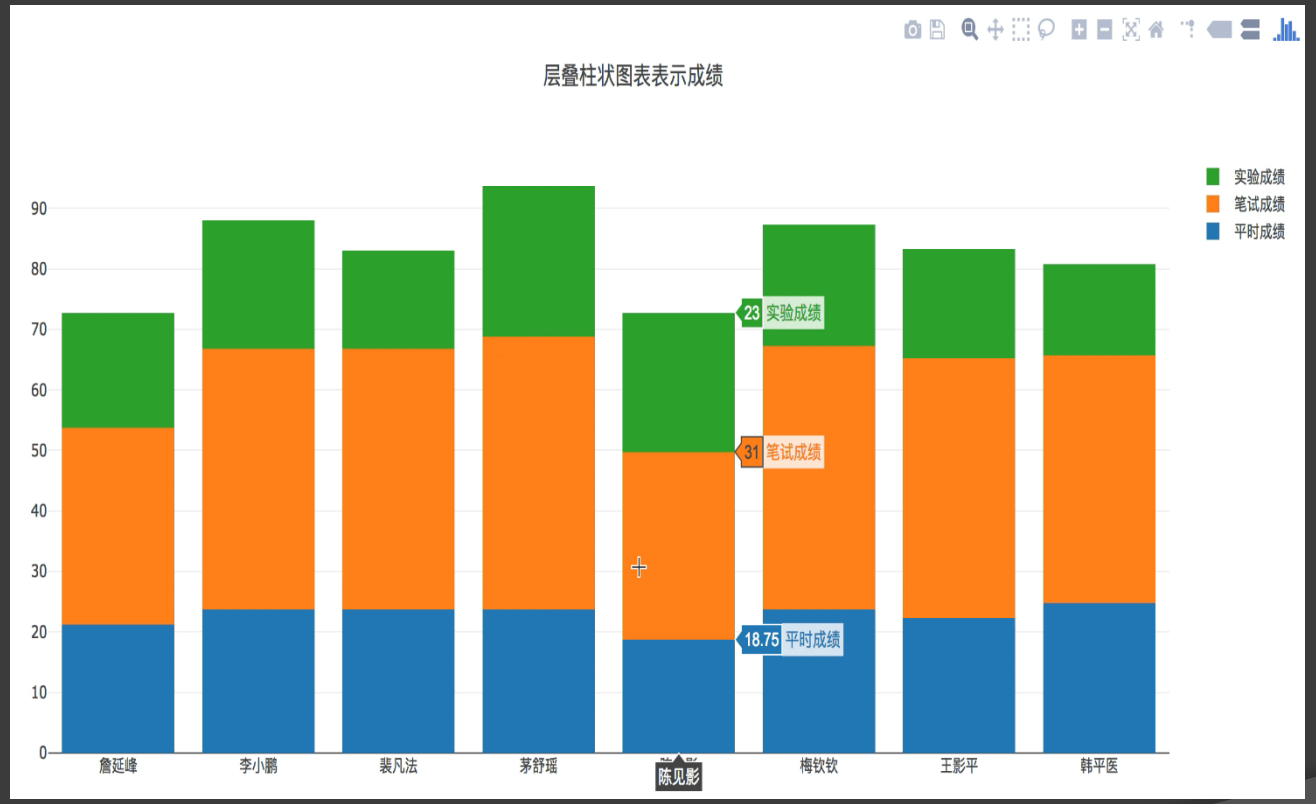
绘制成绩柱状图

```
from plotly.offline import  plot
import plotly.graph_objs as go

import pandas as pd
data = pd.read_csv("score.csv",encoding="GBK")
xdata=data["姓名"].tolist() #取姓名这 一列，变列表
ydata1=(data["平时"]*0.25).tolist() #取平时成绩这 一列，变列表
ydata2=(data["笔试"]*0.5).tolist()
ydata3=(data["实验"]*0.25).tolist()

trace0=go.Bar(x=xdata,y=ydata1,name="平时成绩")
trace1=go.Bar(x=xdata,y=ydata2,name="笔试成绩")
trace2=go.Bar(x=xdata,y=ydata3,name="实验成绩")
layout=go.Layout(title="层叠柱状图表表示成绩",barmode='stack')

fig=go.Figure(data=[trace0,trace1,trace2],layout=layout)
plot(fig)
```

柱状图


层叠柱状图表表示成绩
实验成绩
笔试成绩
平时成绩
80
60
40
20
0
詹延峰
李小鹏
装凡法
茅舒瑶
陈见影
梅钦钦
王影平
韩平医

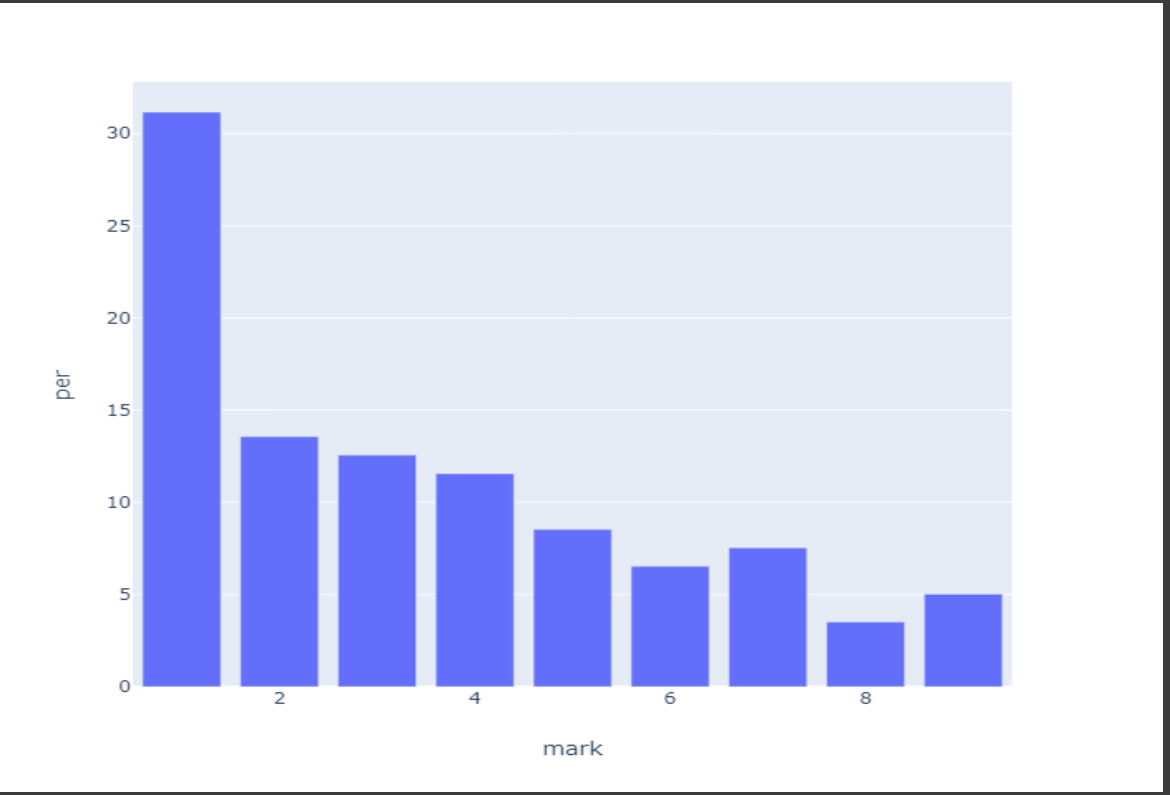
显示各部分成绩


层叠柱状图表表示成绩
实验成绩
笔试成绩
平时成绩
90
80
70
60
50
40
30
20
10
0
23 实验成绩
31 笔试成绩
18.75 平时成绩
詹延峰
李小鹏
裴凡法
茅舒瑶
陈见影
梅钦钦
王影平
韩平医

Benford定律验证

- 美国的一位叫做本福特的物理学家在图书馆翻阅对数表时发现，对数表的头几页比后面的页被更多的人翻阅。
- 本福特再进一步研究后发现，只要数据的样本足够多，数据中以1开头的数字出现的频率并不是1/9，而是30.1%。而以2为首的数字出现的频率是17.6%，往后出现频率依次减少,9的出现频率最低，只有4.6%。
- 本福特开始对其它数字进行调查，发现各种完全不相同的数据，比如人口数据、物理和化学常数、棒球统计表中，均有这个定律的身影。

人口数据的统计结果



验证程序

```
import pandas as pd
from plotly.offline import  plot
import plotly.express as px
digit_counts={d:0 for d in “123456789”}     #用字典解析产生字典
population_file=open('population.txt','r')
population_file.readline() #读标题行
total=0
for line in population_file.readlines():
    line=line.strip().split()
    if line and line[2][0].isdigit():
        first_digit=line[2][0]
        digit_counts[first_digit]+=1
        total+=1
percents=sorted([(int(digit),count*100/total)
                        for digit,count in digit_counts.items()])
dataset=pd.DataFrame(percents,columns=("mark","per"))
fig=px.bar(dataset,x="mark",y="per")
plot(fig)
```

```
rank,national,pop,percent
1  中国 1,405,372,834 18.82%
2  印度 1,304,200,000 17.86%
3  美国 322,760,000 4.42%
4  印尼 257,740,000 3.53%
5  巴西 205,290,000 2.81%
```

可视化程序


```
import pandas as pd
from plotly.offline import  plot
import plotly.express as px

dataset=pd.read_excel("人均GDP和人均寿命1900.xlsx")
figure = px.scatter(dataset, x="income", y="life-exp", animation_frame="year",
                    animation_group="country",size="income", color="continent",
                    hover_name="country",log_x=True, size_max=45,
                    range_x=[500,200000], range_y=[25,90],
                    labels=dict(income="人均收入(PPP购买力标准)",lifeExp="人均寿命"))

plot(figure)
```