# 数据科学实验报告

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 班级 |
| 222018603193119 | 莫健行 | 18级计算机科学与技术3班 |

## 实验题目：

数据可视化

## 实验内容：

1. **Example 1：Show Simple Histogram**
   1. Main problem: Display square of x in histogram.

[Source code]

def data\_test():

# setting x data

data = np.array([i for i in range(10)])

# setting y data

y\_data = data\*data

# use dark style

plt.style.use("dark\_background")

fig, ax = plt.subplots()

ax.set\_title("Square")

ax.bar(data, y\_data)

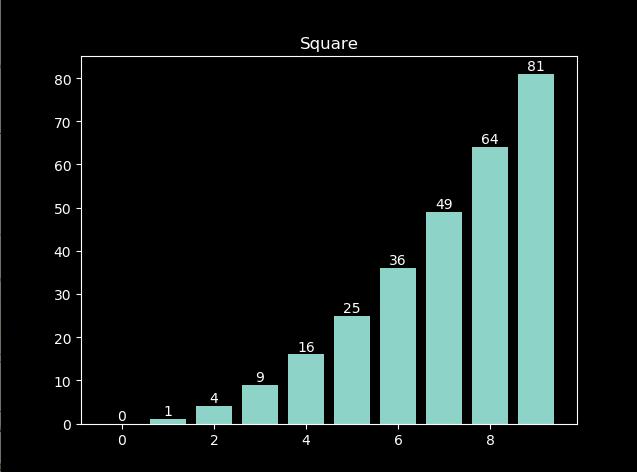
# show the y data on bar

for a,b in zip(data, y\_data):

plt.text(a, b + 0.2, "%d"%b, ha='center', va='bottom', fontsize=10)

plt.show()

[Result]



* 1. Exercise: Change style of gragh

[Source code]

# get available style list

styles = plt.style.available

print(styles, len(styles))

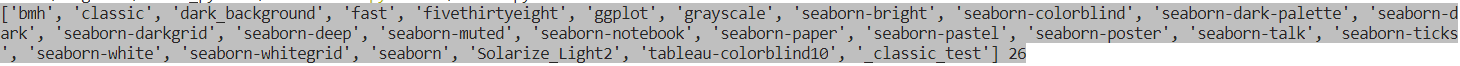
for i in range(9):

# show gragh in style[i]

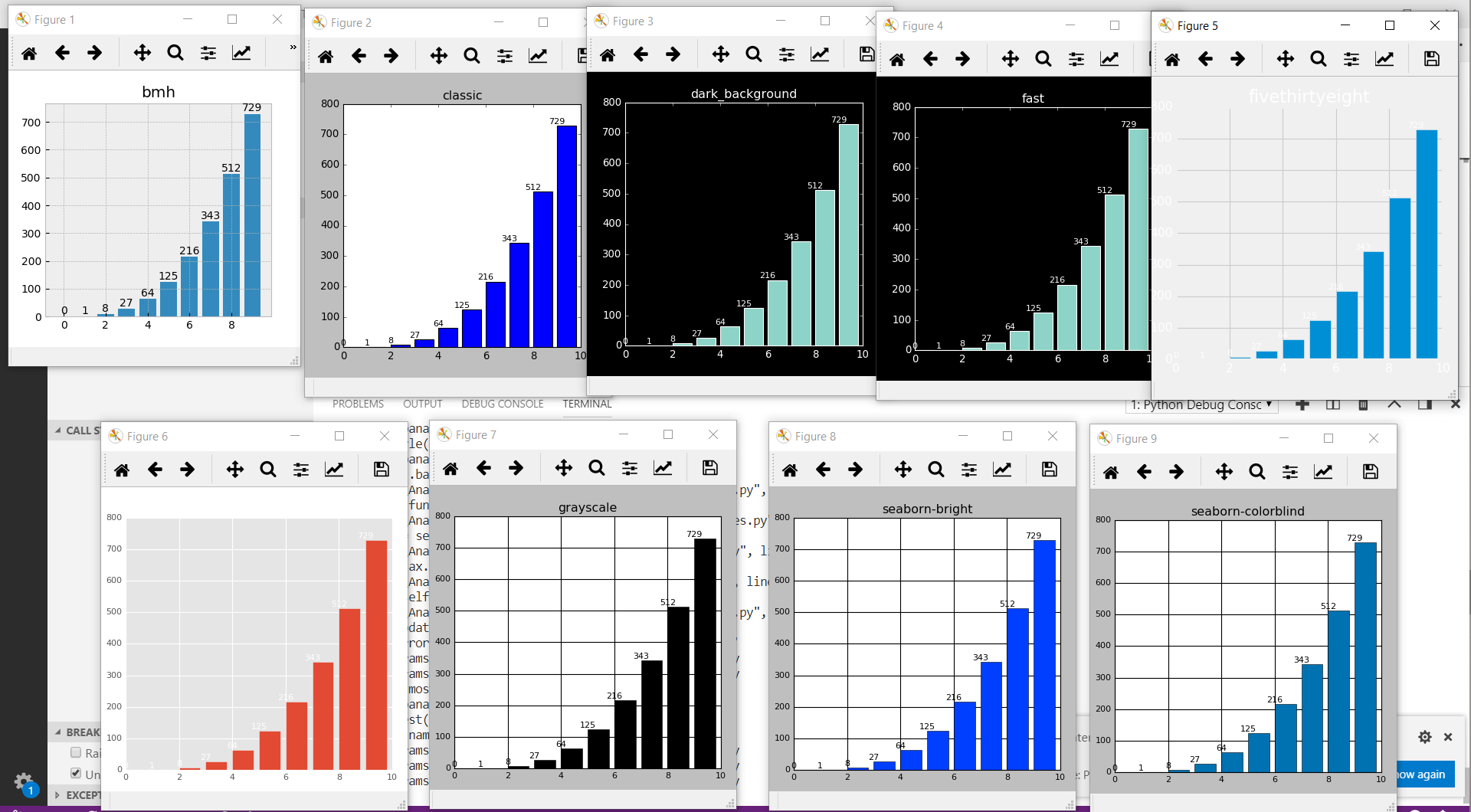
data\_test(styles[i])

plt.show()

[Output]



[Result]



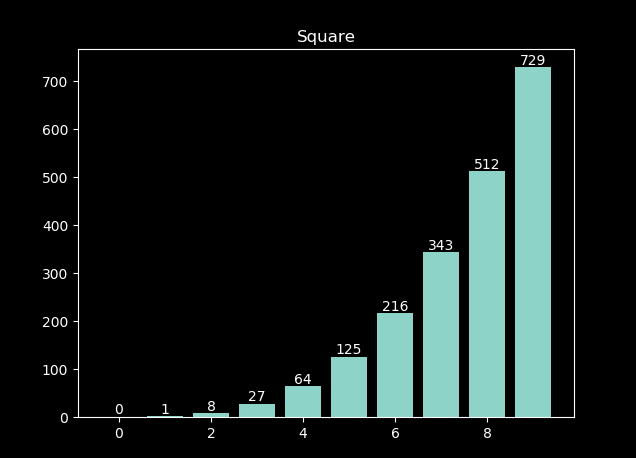
* 1. Exercise: Plot graph of

[Source code]

# use power(x, n) function to calculate x to the power of n

y\_data = np.power(data, 3)

[Result]



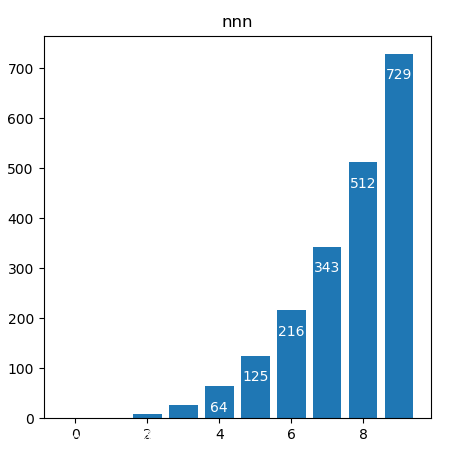
* 1. Change position of digits on histogram

[Source code]

# display text inside bar, and set color to white rgb(1,1,1)

plt.text(a, b-57, "%d"%b, ha='center', va='bottom', fontsize=10, color=(1,1,1))

[Result]



1. **Example 2：Distribution of Data**
   1. Main Problem: Use histogram to show number of student in same score

[Source code]

# read csv file

def loadcsv(path):

data = []

with open(path, 'r') as f:

content = csv.reader(f)

for i in content:

data.append(float(i[0]))

return data

# process data into a dict

def pr\_data():

scores = loadcsv("data.csv")

scores\_data = {}

for i in scores:

scores\_data[int(i)] = scores\_data.get(int(i), 0) + 1

return scores\_data

# show data

def show\_data():

data = pr\_data()

fig = plt.figure()

ax = plt.axes()

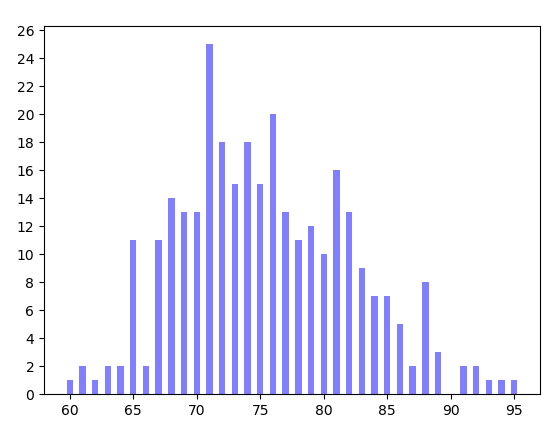
ax.set\_xticks([i for i in range(50, 101, 5)])

ax.set\_yticks([i for i in range(0, 31)])

plt.bar(data.keys(), data.values(), 0.5, alpha=0.5, color='b')

plt.show()

[Result]



* 1. Exercise: Generate some data and instaniate a histogram to show it.

# Generate 100 random number and show

[Source code]

#generate random data

def gen(n):

data = []

for i in range(n):

data.append(random.randint(0, 100))

return data

# show data

def show2():

data = gen(100)

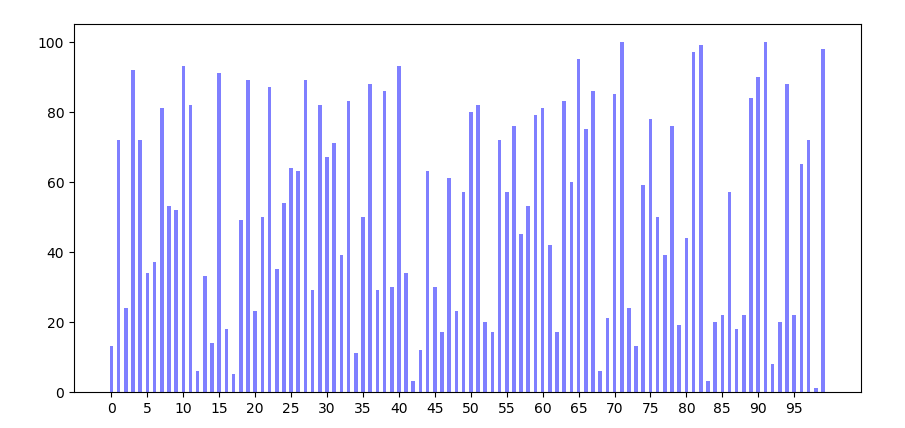
fig, ax = plt.subplots()

ax.set\_xticks(range(0, 100, 5))

ax.bar(range(0, 100), data, 0.5, alpha=0.5, color='b')

plt.show()

[Result]



* 1. Exercise: Requiring that we make every section of 5 point, show count of every section.

[Source code]

# New function to process data, calculate students in every section

def pr\_data\_5():

scores = loadcsv("data.csv")

scores\_data = {}

for i in scores:

scores\_data[int(i)//5] = scores\_data.get(int(i)//5, 0) + 1

return scores\_data

# show data

def show\_data():

data = pr\_data\_5()

fig = plt.figure()

ax = plt.axes()

# x axis

x = [i\*5 for i in data.keys()]

ax.set\_xticks([i for i in range(50, 101, 5)])

# hide origin x axis

ax.xaxis.set\_major\_formatter(plt.NullFormatter())

ax.bar(x, data.values(), 3, alpha=0.5, color='b')

# show the y data on bar

for a,b in zip(x, data.values()):

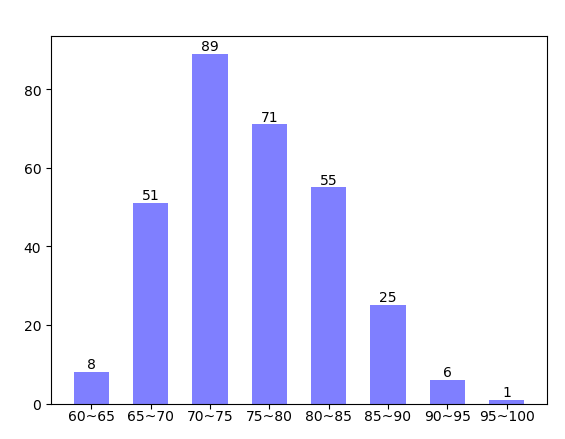
plt.text(a, b + 0.2, "%d"%b, ha='center', va='bottom', fontsize=10)

# show x axis by text

plt.text(a, -5, "%d~%d"%(a, a+5), ha='center', va='bottom', fontsize=10)

plt.show()

[Result]



1. **Example 3: Scatter**
   1. **Example: Categories of Iris**

[Source code]

import csv

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# fig, ax = plt.subplots()

iris = pd.read\_csv("iris.csv",error\_bad\_lines=False)

colors = ['r', 'y','b']

species = iris.Species.unique()

\_, ax = plt.subplots(1,2)

for i in range(len(species)):

ax[0].scatter(iris.loc[iris.Species == species[i], 'Petal.Length'], iris.loc[iris.Species == species[i],'Petal.Width'], s = 35, c = colors[i], label = species[i])

ax[1].scatter(iris.loc[iris.Species == species[i], 'Petal.Width'], iris.loc[iris.Species == species[i],'Petal.Length'], s = 35, c = colors[i], label = species[i])

ax[0].set\_title("Length vs Width")

ax[0].set\_xlabel("Length")

ax[0].set\_ylabel("Width")

ax[0].grid(True, linestyle=":", alpha=0.5)

ax[0].legend(loc="lower right")

ax[1].set\_title("Width vs Length")

ax[1].set\_xlabel("Width")

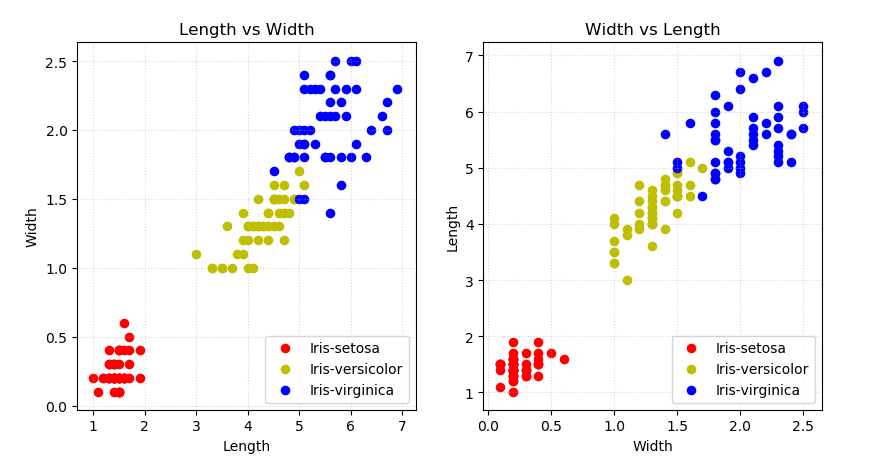
ax[1].set\_ylabel("Length")

ax[1].grid(True, linestyle=":", alpha=0.5)

ax[1].legend(loc="lower right")

plt.show()

[Result]



* 1. **Use iris dataset to draw scatter**

[Source code]

# select two cols from 4 cols

def gen(d):

t = []

cols = d.columns

for i in range(len(cols)-1):

temp = list(cols[:-1])

temp.pop(i)

[t.append([cols[i], j]) for j in temp]

print(t)

return t

iris = pd.read\_csv("iris.csv",error\_bad\_lines=False)

# generate two cols

t = gen(iris)

colors = ['r', 'y','b']

species = iris.Species.unique()

\_, ax = plt.subplots(3,4, figsize=(10,10))

for j in range(12):

plt.subplot(3, 4, j+1)

for i in range(len(species)):

plt.scatter(iris.loc[iris.Species == species[i], t[j][0]], iris.loc[iris.Species == species[i],t[j][1]], s = 35, c = colors[i], label = species[i])

plt.title("{} vs {}".format(t[j][0], t[j][1]))

plt.xlabel(t[j][0])

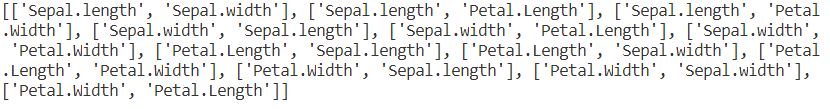
plt.ylabel(t[j][1])

plt.grid(True, linestyle=":", alpha=0.5)

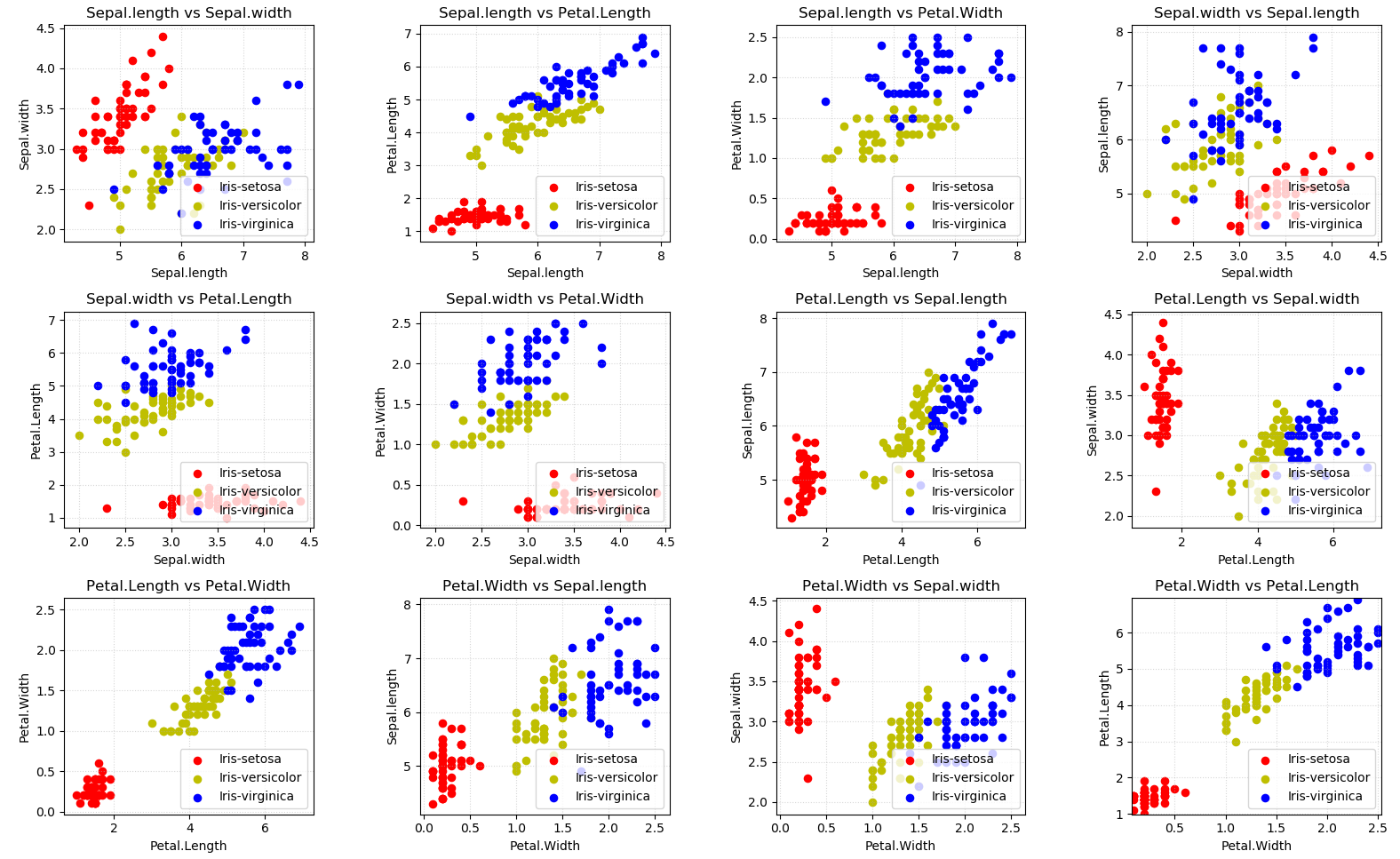
plt.legend(loc="lower right")

plt.show()

[Output]



[Result]



1. **Example 4: Changes of Ranks**
   1. Main problem: Draw two group of histogram to show ranks changes between two terms.

[Source code]

# load data

def loadcsv\_2(path):

data = []

with open(path, 'r') as f:

content = csv.reader(f)

for i in content:

data.append((int(i[0]), int(i[1])))

return data

# show data

def show3():

data = loadcsv\_2("data2.csv")

fig, ax = plt.subplots()

x = np.array([i+1 for i in range(len(data))])

ax.set\_xticks(range(1,11))

# x-0.2 and x+0.2 make two bar stand on sides of value

ax.bar(x-0.2, [i[0] for i in data], 0.4, alpha=0.5, color='r')

ax.bar(x+0.2, [i[1] for i in data], 0.4, alpha=0.5, color='b')

for a in x:

a = a-1

ax.text(a - 0.2 + 1, data[a][0], "%d"%data[a][0], ha='center', va='bottom', fontsize=10)

ax.text(a + 0.2 + 1, data[a][1], "%d"%data[a][1], ha='center', va='bottom', fontsize=10)

ax.legend(["This Term","Last Term"], loc='upper left')

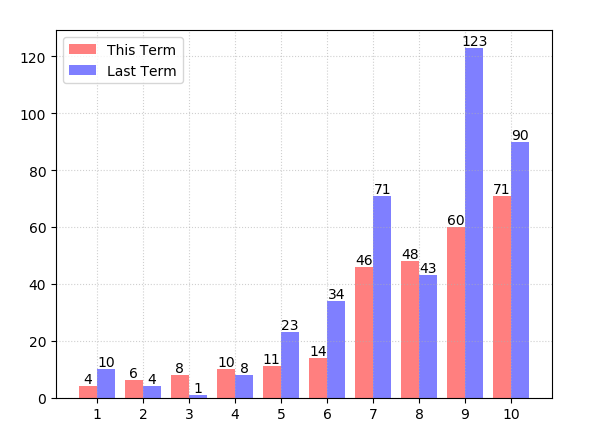
# set grid

ax.grid(True, linestyle=':', alpha=0.6)

plt.show()

[Result]

Place bars on two side of vertical base line



* 1. Exercise: Show data in 3 terms

[Source code]

# generate data of third term

def gen2(d):

data = []

for i in d:

t = i[1] + random.randint(-30, 30)

# judge whether reach border

t = 1 if t < 1 else 200 if t > 200 else t

data.append((i[0], i[1], t))

return data

# show data

def show4():

data = gen2(loadcsv\_2("data2.csv"))

fig, ax = plt.subplots()

x = np.array([i+1 for i in range(len(data))])

ax.set\_xticks(range(1,11))

# x-0.2 and x+0.2 make two bar stand on sides of value

ax.bar(x-0.2, [i[0] for i in data], 0.2, alpha=0.5, color='r')

ax.bar(x, [i[1] for i in data], 0.2, alpha=0.5, color='g')

ax.bar(x+0.2, [i[2] for i in data], 0.2, alpha=0.5, color='b')

for a in x:

a = a-1

ax.text(a - 0.2 + 1, data[a][0] + 0.2, "%d"%data[a][0], ha='center', va='bottom', fontsize=10)

ax.text(a + 1, data[a][1] + 0.2, "%d"%data[a][1], ha='center', va='bottom', fontsize=10)

ax.text(a + 0.2 + 1, data[a][2] + 0.2, "%d"%data[a][2], ha='center', va='bottom', fontsize=10)

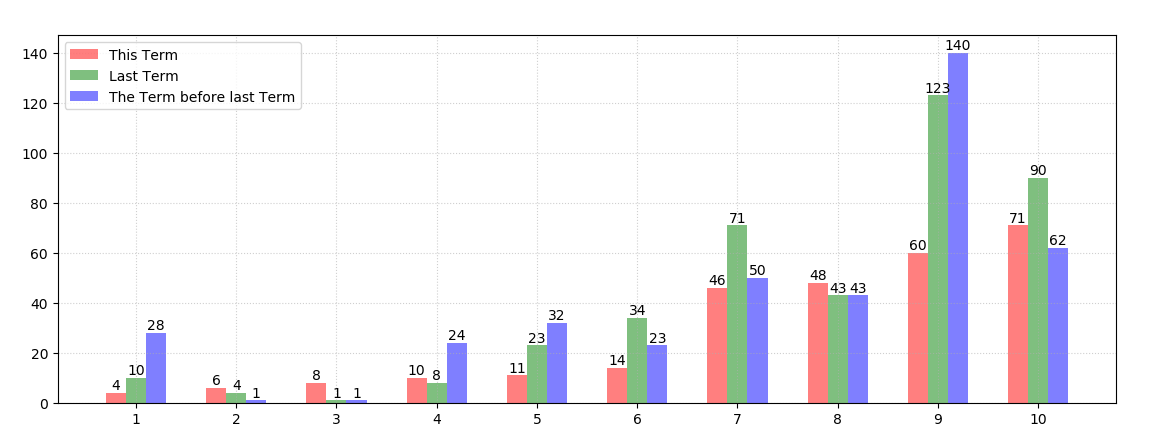
ax.legend(["This Term","Last Term", "The Term before last Term"], loc='upper left')

# set grid

ax.grid(True, linestyle=':', alpha=0.6)

plt.show()

[Result]



1. Data Visualization —— Map
   1. Example 1: Show amount of universities in each province of China

[Source code]

from pyecharts import options as opts

from pyecharts.charts import Map

import random

class Data:

provinces = ["湖北", "广东", "北京", "上海", "江西", "河南", "浙江", "江苏",

"湖南", "广西", "山东", "陕西", "山西", "河北", "福建", "黑龙江",

"新疆", "西藏", "云南", "贵州", "四川", "台湾", "宁夏", "吉林",

"青海", "甘肃", "内蒙古", "重庆", "安徽","天津","海南","辽宁"]

@staticmethod

def values(start:int = 20, end:int = 150)->list:

return [random.randint(start, end) for i in range(32)]

def mp() -> Map:

c = (

Map()

.add("各省数量", [list(i) for i in zip(Data.provinces, Data.values())], "china")

.set\_global\_opts(

title\_opts = opts.TitleOpts(title="Map"),

visualmap\_opts = opts.VisualMapOpts(min\_=20, max\_=150)

)

.set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=True))

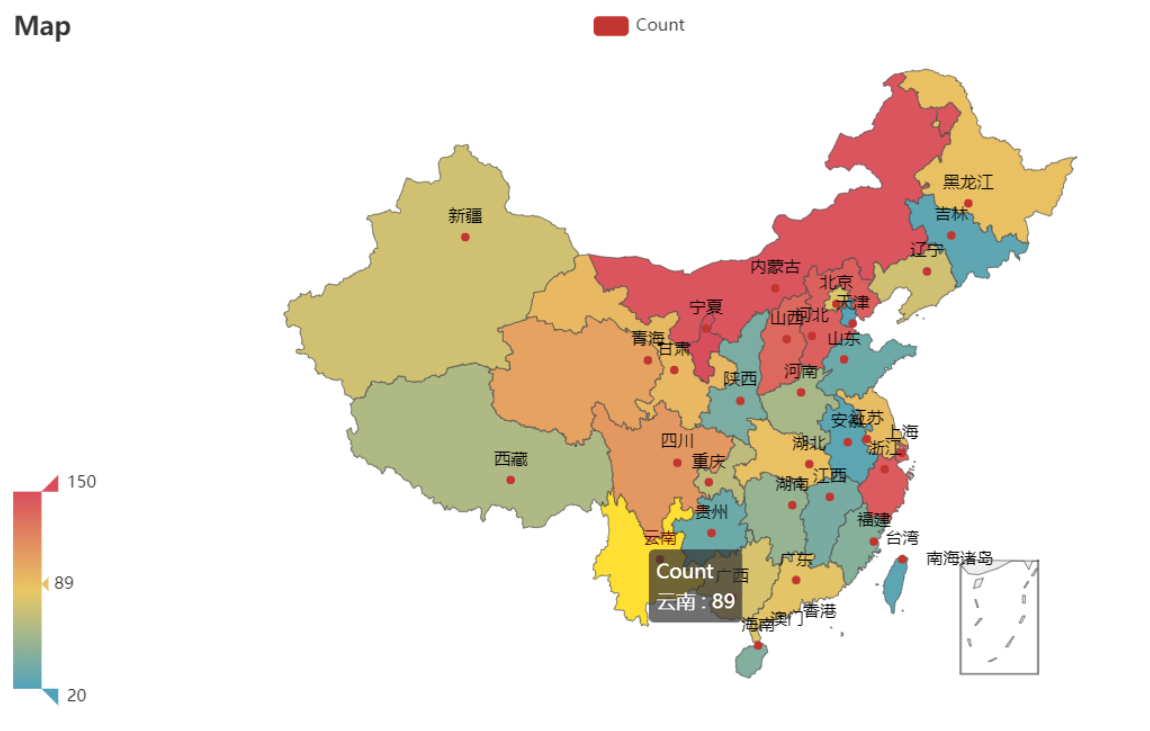
)

return c

mp().render("map.html")

[Result]

map.html



* 1. Exercise: Show student amount that attend UEE(University Enchance Examination) each province

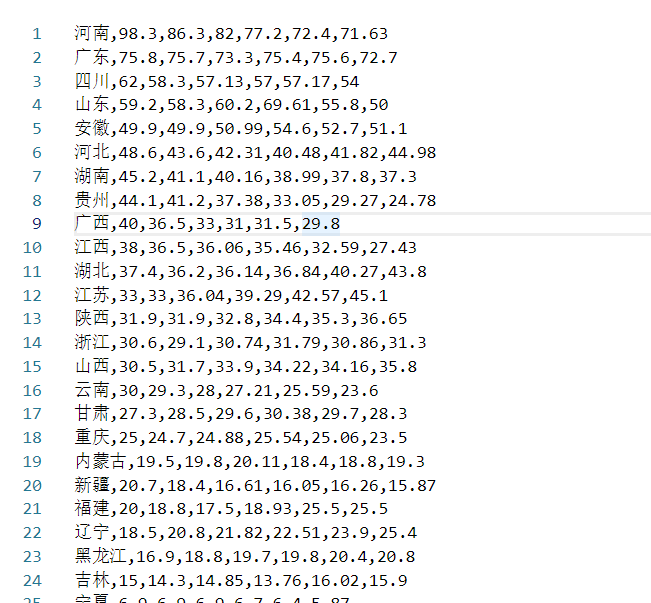
[Source code]

# 可以显示三年的数据

1. Step 1: get data from internet



1. Step 2: save as csv



1. Step 3: load csv data

def loadcsv():

provinces = []

data = {2018: [], 2017: [], 2016: []}

with open("datata.csv",'r') as f:

content = csv.reader(f)

for i in content:

provinces.append(i[0])

data[2018].append(float(i[1]))

data[2017].append(float(i[2]))

data[2016].append(float(i[3]))

return provinces, data

p, d = loadcsv()

1. Step 4: build data class and

class Data1:

global p

global d

provinces = p

@staticmethod

def get\_values(y:int)->list:

return d[y]

1. add data in recent 3 years to Map()

def mp() -> Map:

c = (

Map()

.add("2018", [list(i) for i in zip(Data1.provinces, Data1.get\_values(2018))], "china")

.add("2017", [list(i) for i in zip(Data1.provinces, Data1.get\_values(2017))], "china")

.add("2016", [list(i) for i in zip(Data1.provinces, Data1.get\_values(2016))], "china")

.set\_global\_opts(

title\_opts = opts.TitleOpts(title="Map"),

visualmap\_opts = opts.VisualMapOpts(min\_=0, max\_=100)

)

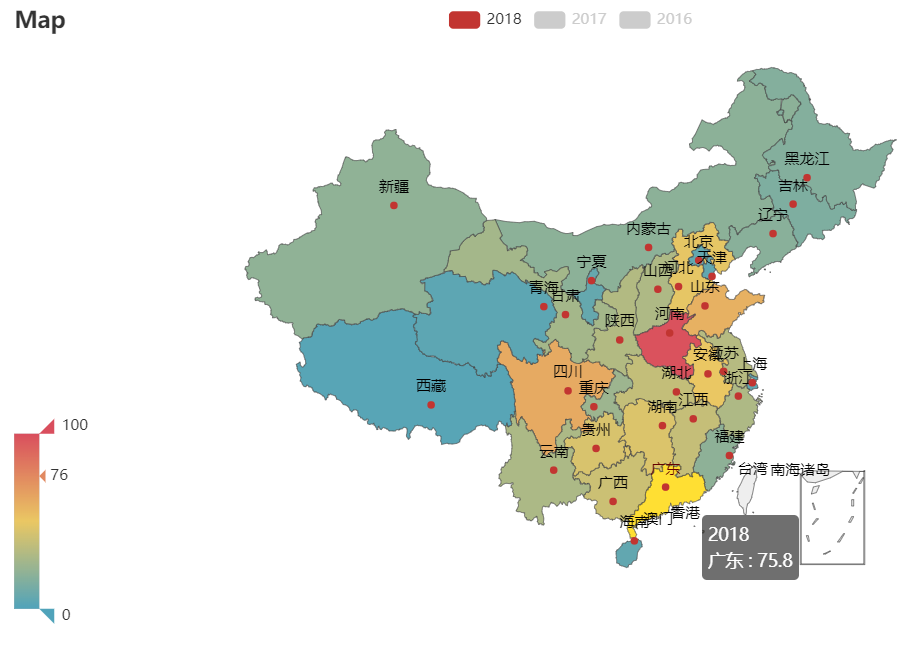
.set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=True))

)

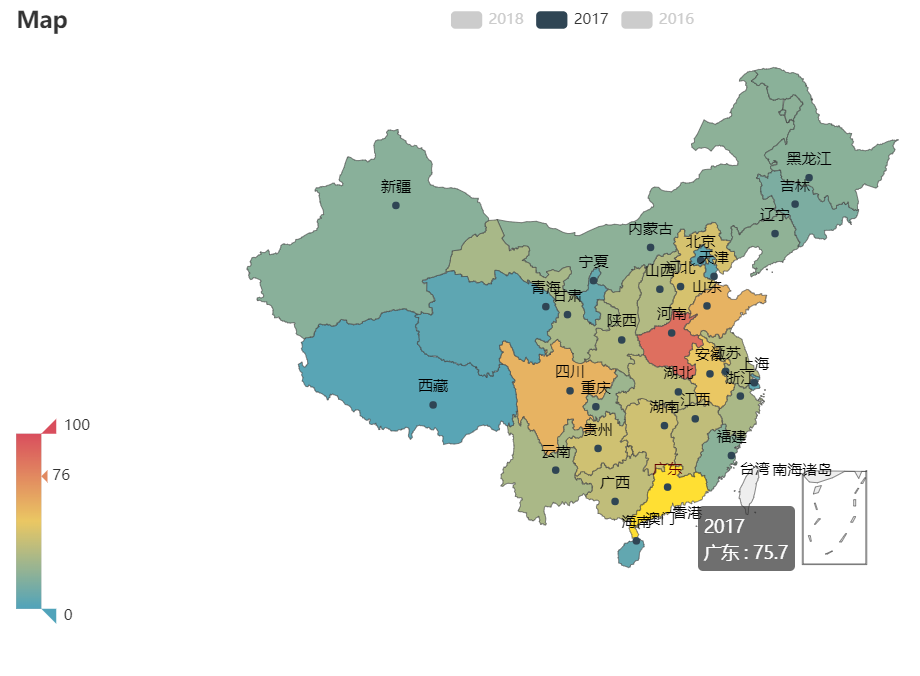
return c

[Result]

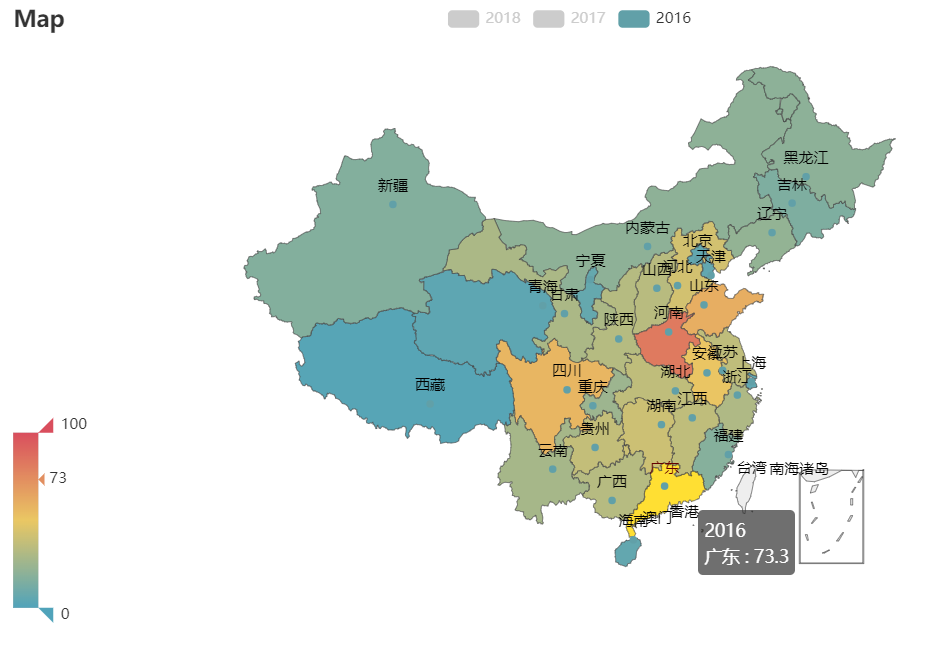
**Data in 2018**



**Data in 2017**



**Data in 2016**



* 1. Example 2: Show the heat map

[Source code]

class Data:

guangdong\_city = ["佛山市","湛江市","潮州市","河源市","江门市", "中山市", "珠海市", "深圳市","东莞市","韶关市", "清远市", "云浮市", "茂名市","汕头市", "汕尾市", "揭阳市", "阳江市", "肇庆市", "广州市", "惠州市","梅州市"]

@staticmethod

def values(start:int = 10, end:int = 40)->list:

return [random.randint(start, end) for i in range(21)]

def heat(date)->Geo:

return (

Geo()

.add\_schema(maptype="广东")

.add(

"8月", [list(i) for i in zip(Data.guangdong\_city, Data.values())], type\_=ChartType.HEATMAP

)

.set\_global\_opts(

visualmap\_opts=opts.VisualMapOpts(max\_=42, is\_piecewise=True),

title\_opts=opts.TitleOpts(title="广东"+date+"各地区温度变化情况")

)

)

path = "8月{0}日"

for i in range(5):

#heat().render(path.format(i+1))+ ".html")

make\_snapshot(snapshot, heat(path.format(i+1)).render(), path.format(i+1)+".png", pixel\_ratio=1)

# Genrate gif

[Result]

