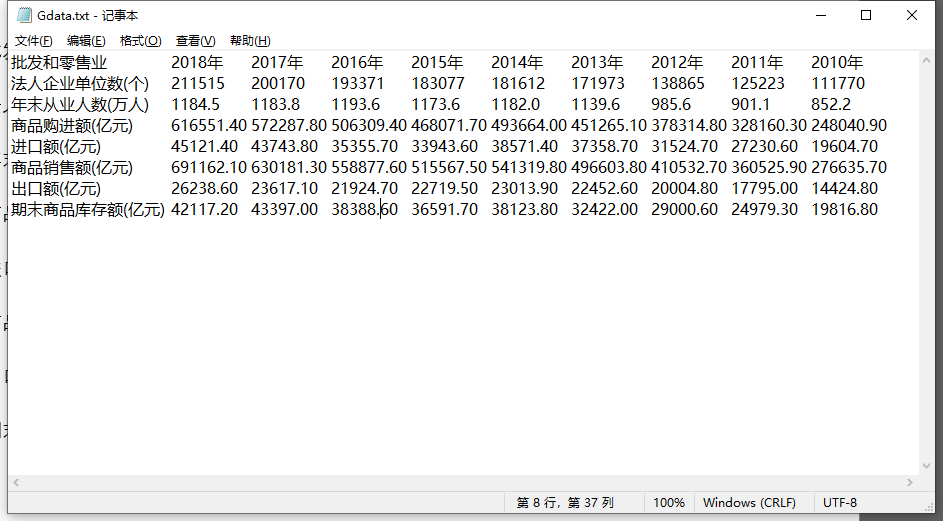
数据介绍：

批发和零售业的各项数据，老师课上好像没有设计到CVS格式，故我采用txt文件格式。由于每个项目名字太长，就将批发和零售业单独提出来，放到“项目”处替换“项目”。（如不符合健壮性也可以随时变更）注意运行文件时候更改文件的路径



批发和零售业 2018年 2017年 2016年 2015年 2014年 2013年 2012年 2011年 2010年

法人企业单位数(个) 211515 200170 193371 183077 181612 171973 138865 125223 111770

年末从业人数(万人) 1184.5 1183.8 1193.6 1173.6 1182.0 1139.6 985.6 901.1 852.2

商品购进额(亿元) 616551.40 572287.80 506309.40 468071.70 493664.00 451265.10 378314.80 328160.30 248040.90

进口额(亿元) 45121.40 43743.80 35355.70 33943.60 38571.40 37358.70 31524.70 27230.60 19604.70

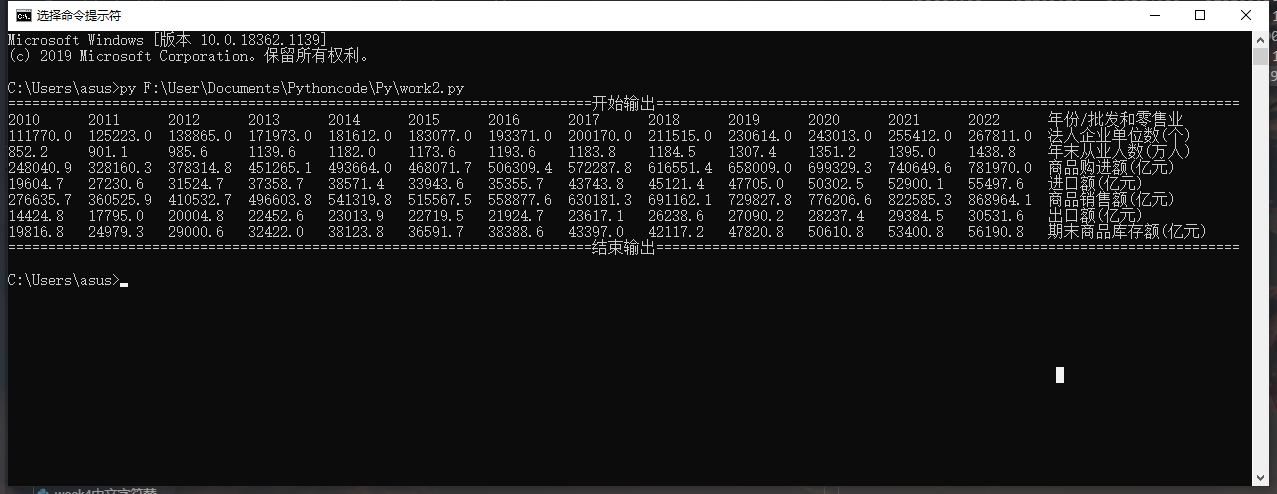
商品销售额(亿元) 691162.10 630181.30 558877.60 515567.50 541319.80 496603.80 410532.70 360525.90 276635.70

出口额(亿元) 26238.60 23617.10 21924.70 22719.50 23013.90 22452.60 20004.80 17795.00 14424.80

期末商品库存额(亿元) 42117.20 43397.00 38388.60 36591.70 38123.80 32422.00 29000.60 24979.30 19816.80

结果展示：

数据仅有2010至2018年的数据，因此处于预测和验证考虑，将2019至2022年的数据算出来。因没有近两年的数据，故无法对比。



我将项目名称移动至最右，否则数据会不对齐影响观赏性。

源代码：

1. def means(List):
2. **return** sum(List)/len(List)
4. def linearRegression(xlist,ylist):
5. xmean,ymean = means(xlist),means(ylist)
6. bNumerator = -len(xlist)\*xmean\*ymean
7. bDenominator = -len(ylist)\*xmean\*\*2
8. **for** x , y in zip(xlist,ylist):
9. bNumerator += x\*y
10. bDenominator += x\*\*2
11. b = bNumerator/bDenominator
12. a = ymean - b\*xmean
13. **return** b,a
15. def Datacollect():
16. dataname , data = [] , []
17. with open("F:\\User\\Documents\\Pythoncode\\data\\Gdata.txt", mode='r' , encoding = "utf-8") as f:
18. **for** line in f:
19. splitedLine = line.strip().split()
20. **if** '批发和零售业' in splitedLine[0]:
21. years = [**int**(x[:-1]) **for** x in splitedLine[1:]]
22. **else**:
23. dataname.append('{:<10}'.format(splitedLine[0]))
24. data.append([**float**(x) **for** x in splitedLine[1:]])
25. **for** i in range(len(dataname)):
26. **for** ch in " ":
27. dataname[i] = dataname[i].replace(ch,"")
29. **return** years,dataname,data
31. def NumSolve(slope,intercept):
32. **return** slope\*2019+intercept,slope\*2020+intercept,slope\*2021+intercept,slope\*2022+intercept
34. Year,Dataname,Data=Datacollect()
35. Year.reverse()
36. **for** i in range(len(Data)):
37. Data[i].reverse()
39. **for** i in range(len(Data)):
40. bData,aData = linearRegression(Year,Data[i])
41. newdata = NumSolve(bData,aData)
42. **for** j in range(len(newdata)):
43. Data[i].append(newdata[j])
45. **for** i in [2019,2020,2021,2022]:
46. Year.append(i)
48. Dataname.insert(0,"年份/批发和零售业")
49. Data.insert(0,Year)
50. **for** i in range(1,len(Data)):
51. **for** j in range(len(Data[0])):
52. **if** i < 2:
53. Data[i][j] = round(Data[i][j],0)
54. elif i == 3:
55. Data[i][j] = round(Data[i][j],1)
56. **else**:
57. Data[i][j] = round(Data[i][j],1)
58. print("{:=^150}".format("开始输出"))
59. **for** i in range(len(Dataname)):
60. **for** j in range(len(Data[0])):
61. print("{:<10}".format(Data[i][j]), end= '')
62. print("{:<15}\n".format(Dataname[i]),end = '')
63. print("{:=^150}".format("结束输出"))