

分析拥有3-5年工作经验python工程师的薪资与学历的关系

——数据挖掘与数据分析

学号姓名：16103218魏 远

16103208李东升

2019年6月5日

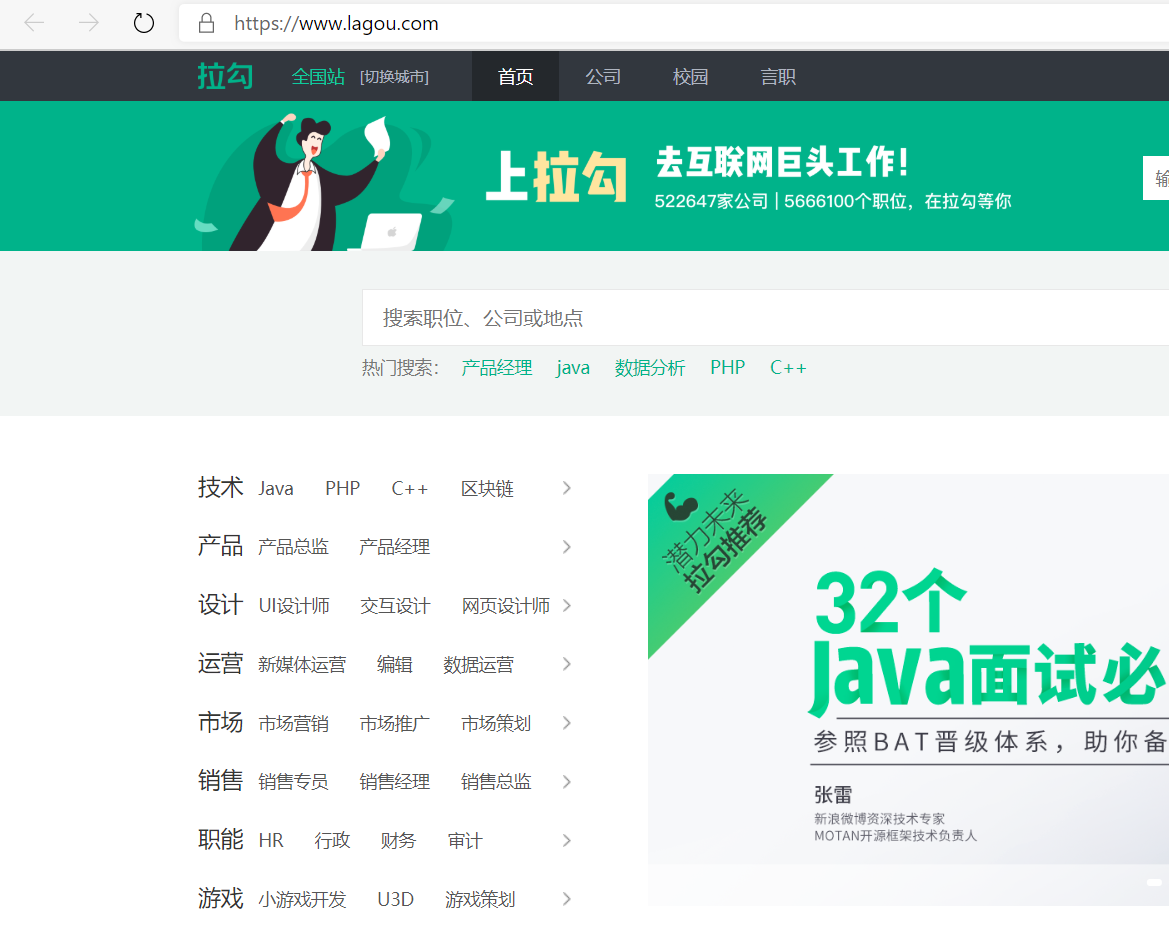
问题说明

在近年的就业市场中，就业形式越发严峻。特别是近年来一二本合并，本科生扩招，直接增加了应届毕业生的压力，所以不少应届生选择深造，且不论深造的结果成功与否，但越来越多的高学历人才涌现，也使得企业对应届生的要求也逐年升高，而因千禧年后计在算机领域焕发生机的中国产生的计算机相关的毕业生的红利却在渐渐殆尽，那么对近年来本科应届生的求职就增加了考验。越来越多的本科应届生陷入了困惑，我的学历足够支撑我未来想要的薪资吗？如果能达到，需要花费几年？还是说应该深造，将目标放长远一些会有更好的效果？为了回答这个问题，本小组将基于对招聘网站的数据，分析薪资与学历的关系，给近年的应届生做一个参考。

数据来源

知名招聘网站：拉勾网

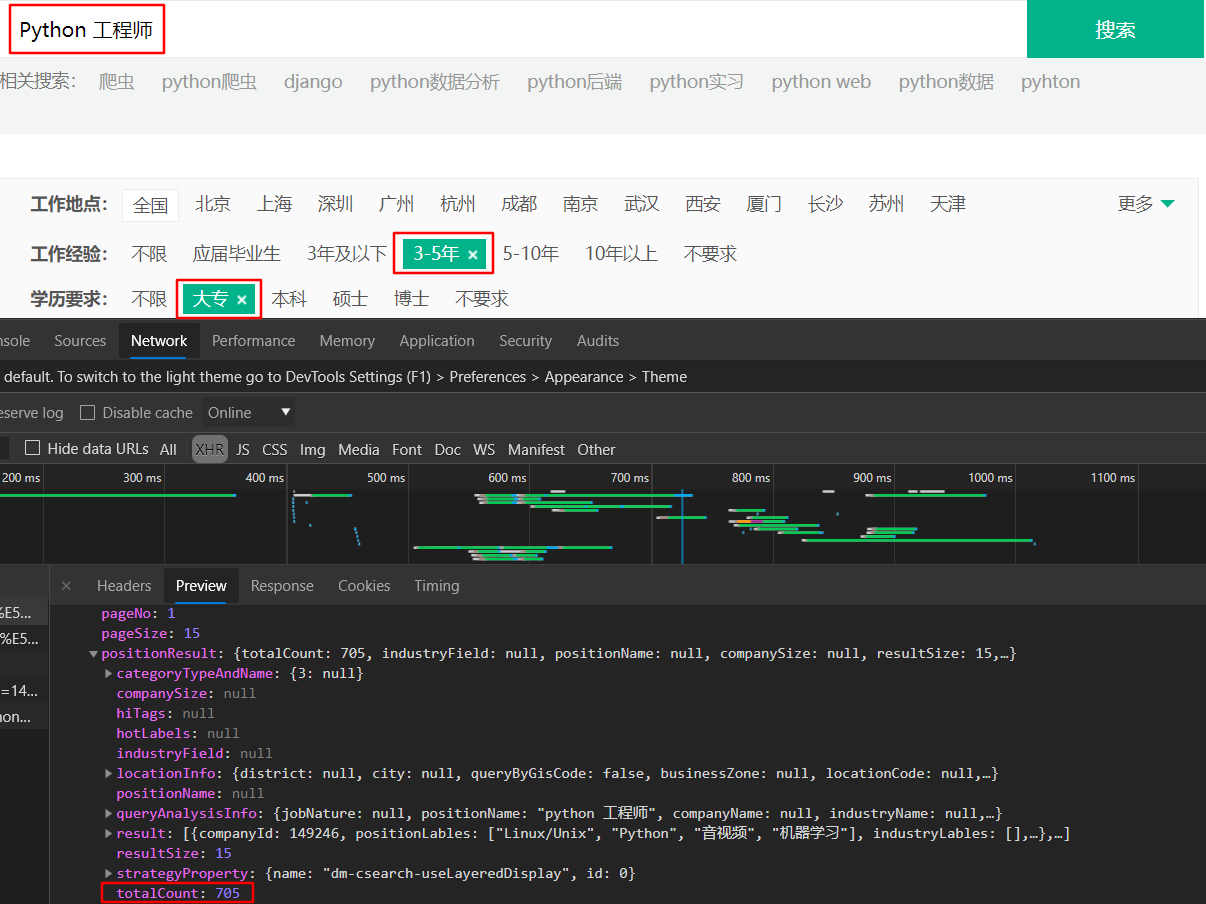
网址：<https://www.lagou.com/>



预处理过程

1：数据限定条件的限制

为了让结果更具代表性，我们小组将职业限定为近年比较火爆的“Python工程师”，选择工作经验为3-5年，这里没有将工作经验选为“应届生”的原因是工作经验为“应届生”的数据，不论是专科，本科，还是硕士，都良莠不齐，不论是数量上还是质量上都难有保障。所以我们选择了数据量充足且有可比性的工作经验为3-5年的Python工程师的薪资作为数据。而且将工作经验设置为3-5年还能观察不同学历的求职者在工作年限稳定之后的薪资水平。这能使得本题的分析有更说服力。



2：数据的转换

我们爬下来的薪资数据是以形如 "salary": "13k-20k" 的键值对形式展示出来的，但是值数据地 “13k-20k” 是一个字符串不利于我们进行分析，所以我们在画直方图地时候用了正则表达式进行数据的转换，将 "salary": "13k-20k" 转换为两个int 数据方面传入数据，具体代码下面数据分析阶段会讲解。

挖掘方法原理

拉勾网是一个动态网页。传统的rs + bs已经不再适合。在这里我们采用selenium直接模拟浏览器的方式。模拟 chrome 浏览器打开拉勾网。这样只要是我们能够看到的东西都可以爬取下来。下面会对代码进行展示及解析

分析数据的过程中，我们用到了pandas , numpy , matplotlib , 正则表达式等工具包，用于制图及类型转换，还有调用这些工具包提供的函数求出方差，标准差等其他数值，进行类型转换及传入需要的类型。

名词解释：

rs : requests工具包，用于爬取数据时请求URL

bs : beautifulsoup 工具，用于解析网页源码

具体过程及结果展示

**数据获取：**

1：导入的包

1. **import** time
2. **import** json
3. **from** selenium.common.exceptions **import** TimeoutException
4. **from** selenium **import** webdriver
5. **from** selenium.webdriver.common.by **import** By
6. **from** selenium.webdriver.common.keys **import** Keys
7. **from** selenium.webdriver.support **import** expected\_conditions as EC
8. **from** selenium.webdriver.support.wait **import** WebDriverWait
9. **from** urllib.parse **import** quote
10. **from** pyquery **import** PyQuery as pq

具体可以分为 import json, import time, import selenium下的各个包。

1. Json用于数据保存为json格式。
2. time用于爬取数据用到的间隔时间，过一段时间再爬取下一页。防止ip被封。
3. selenium 用于模拟浏览器用地的一些工具，例如等待一个节点可点击，超时异常
4. pyquery 用于分析网页源码，提取出我们需要的数据。
5. quoto 用于url 的链接。链接中文到url里面

2：模拟浏览器函数

1. **def** index\_page(job,educate,page,jobyear):
2. # 先爬取第一页再爬取后面的一直点下一页利用for循环
3. **try**:
4. **print**("正在爬取第1页")
5. url = "https://www.lagou.com/jobs/list\_" + quote(job) + "?px=default" + "&gj=" + quote(jobyear) + "&xl=" + quote(educate)
6. browser.get(url)
7. get\_information(educate)
8. # 先爬取前三页实验一下
9. **for** i **in** range(2,page+1):
10. **print**("正在爬取第{}页".format(i))
11. #等待下一页可以点击的时候再点击
12. submit = wait.until(
13. EC.element\_to\_be\_clickable((By.CSS\_SELECTOR,'div.item\_con\_pa ger div.pager\_container > span.pager\_next'))
14. )
15. submit.click()
16. time.sleep(8)
17. # 等待数据的节点显示出来再调用get\_information()函数
18. get\_information(educate)
19. **except** TimeoutException: # 捕获超出事件异常，从头开始爬取，后续改进
20. index\_page(job=job,educate=educate,page=page,jobyear=jobyear)
21. browser(url) 函数打开的是网站的第一页数据，因为拉钩网采用post方式的第一页和第二页之间的url没有关联。(不能够通过get(url) 的方式获取任意页的数据)所以我们要等待网页下面的下一页是clickable 状态点击它获取下一页的数据
22. 调用get\_information( ) 函数，get\_information( ) 函数用于提取数据。

3：分析网页节点，提取数据

1. **def** get\_information(educate):
2. # 获得数据，公司名和薪资...或者其他的信息
3. html = browser.page\_source
4. # 利用 pyquery CSS选择器
5. doc = pq(html)
6. items = doc("div#s\_position\_list .item\_con\_list li").items()
7. **for** item **in** items:
8. position = item.find("div.position .position\_link span.add em").text()
9. information = {
10. "companyName":item.attr("data-company"),
11. "salary":item.attr("data-salary"),
12. "position":position
13. }
14. **print**(information)
15. infos.append(information)
16. jsObj = json.dumps(infos,ensure\_ascii=False,indent=4)
17. fileObject = open('file.json'.format(educate),'w')
18. fileObject.write(jsObj)
19. fileObject.close()
20. 获取网页代码，html = browser.page\_sourse
21. 利用CSS选择器找到各个数据的节点，并调用 items( ) 方法作为一个列表用于for 循环，循环获取任意一条招聘信息的数据
22. 分析节点数据，将公司名称，薪资学历提取出来。将数据保存到字典里面。然后再将加入到列表里面，供后面保存到json文件里面做准备。
23. 将列表数据保存到json文件里

**数据分析：**

1：保存数据到为csv格式：

1. **import** csv
2. **import** json
3. **import** pandas as pd
5. educates = ["大专","本科","硕士"]
6. **for** i **in** educates:
7. job\_results = []
8. with open("{}.json".format(i),"r") as jf:
9. results = json.load(jf)
10. **for** result **in** results:
11. job\_result = []
12. job\_result.append(result["companyName"])
13. job\_result.append(result["salary"])
14. job\_result.append(result["position"])
15. job\_results.append(job\_result)
16. df = pd.DataFrame(data=job\_results,
17. columns=['公司全名','薪资','区域'])
18. df.to\_csv("{}.csv".format(i),index=False,encoding="utf-8")
19. 导入csv, pandas, json等工具包
20. for循环迭代 “大专，本科，硕士“ 的json文件提取出字典的值，并将值保存到列表里，供后面的pandas.DataFrame( ) 使用
21. 调用 pandas.DataFrame( ) 函数传入列表，提供列名。

2：制出工资频数直方图

1. #coding=utf-8
2. **import** pandas as pd
3. **from** matplotlib **import** pyplot as plt
5. educates = ["大专","本科","硕士"]
7. **for** educate **in** educates:
8. df = pd.read\_csv('{}.csv'.format(educate),encoding="utf-8")
9. pattern = '\d+'
10. df["salary"] = df["薪资"].str.findall(pattern)
12. avg\_salary = []
13. **for** k **in** df["salary"]:
14. int\_list = [int(n) **for** n **in** k]
15. avg\_wage = int\_list[0] + (int\_list[1] - int\_list[0])/4
16. avg\_salary.append(avg\_wage)
17. df["月工资"] = avg\_salary
18. **print**(df["月工资"])
19. **print**(type(df["月工资"]))
21. plt.hist(df["月工资"])
22. plt.xlabel("salary monthly()")
23. plt.ylabel("frequency")
24. plt.title("salary")
25. plt.savefig("{}.jpg".format(educate))
26. plt.show()
27. 导入pandas，matplotlib制表用到的工具。
28. 使用pandas提取出csv 里的数据，并对salary的值进行数据转换，使用正则表达式 ”\d+” 提取出里面的数字。
29. 将数据存入列表里并赋值给df(“月工资”)
30. 绘制工资频数直方图，横坐标为工资，纵坐标表示工资的频数

**3： 保存每条薪资的平均值到json**

1. **import** re
2. **import** json
4. pattern = "\d+"
5. educates = ["大专","本科","硕士"]
6. **def** index(educate):
7. salary\_avgs = []
8. with open("{}.json".format(educate),"r") as jf:
9. salaryavg\_list = []
10. results = json.load(jf)
11. **for** result **in** results:
12. salary = result["salary"]
13. salary1 = re.findall(pattern=pattern,string=salary)
14. **print**(salary1)
15. salary\_avg = int((int(salary1[0]) + int(salary1[1]))/2)
16. salaryavg\_list.append(salary\_avg)
17. salary\_avgs += salaryavg\_list
18. **return** salary\_avgs
20. **def** save\_avg(list,educate):
21. jsObj = json.dumps(list,ensure\_ascii=False)
22. fileObject = open('{}平均.json'.format(educate), 'w')
23. fileObject.write(jsObj)
24. fileObject.close()

27. **if** \_\_name\_\_ == '\_\_main\_\_':
28. **for** educate **in** educates:
29. list = index(educate=educate)
30. save\_avg(list=list,educate=educate)

1：首先加载json文件数据，提取三个学历出每条薪资平均值保存到list 列表里

2：保存list到对应的json文件里

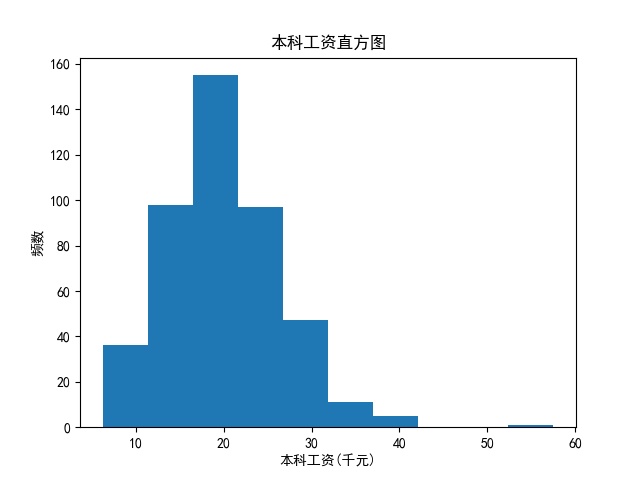
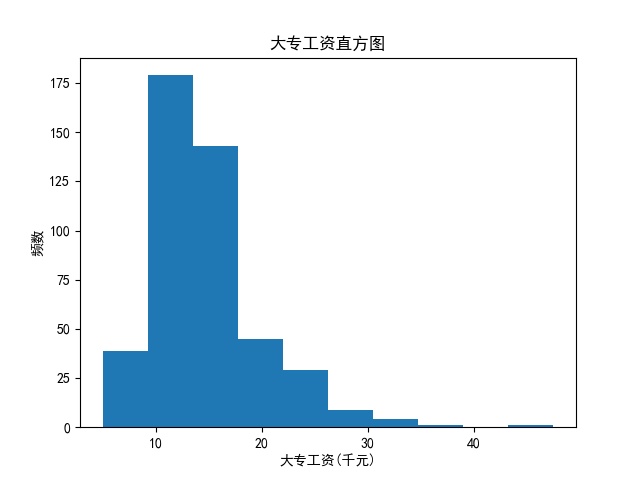
3：这里我们选择保存文件，是为以后绘制其他图做准备

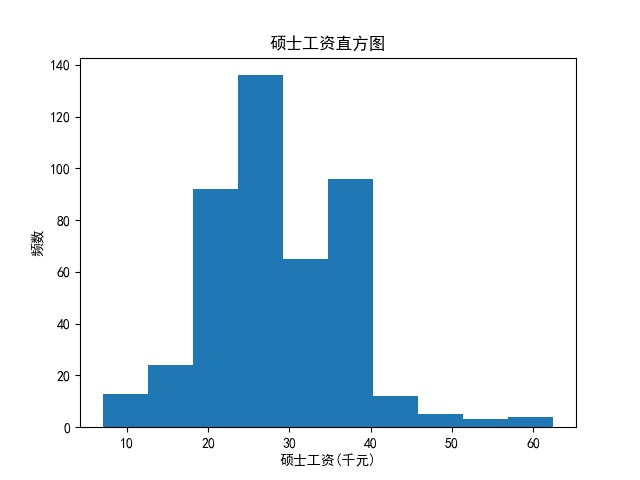
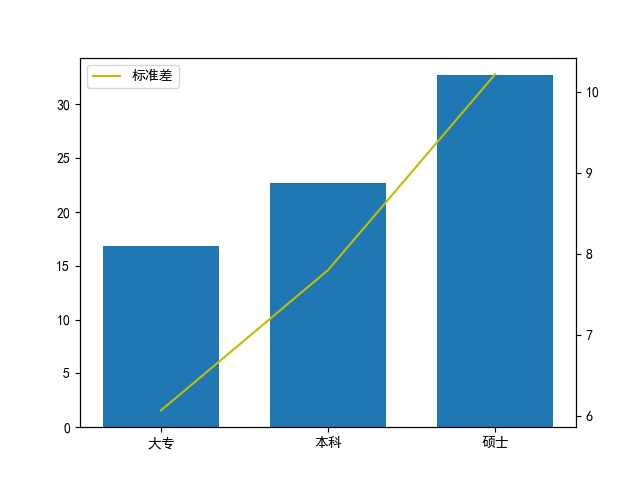
**4：在一个图里绘制薪资平均图和标准差**

1. **import** json
2. **import** numpy as np
3. **from** matplotlib **import** pyplot as plt
5. educates = ["大专","本科","硕士"]
7. **def** average(educate):
8. with open("{}平均.json".format(educate),"r") as jf:
9. result = json.load(jf)
10. **print**(result)
11. arr\_mean = np.mean(result)
12. **return** arr\_mean
14. **def** standard(educate):
15. with open("{}平均.json".format(educate),"r") as jf:
16. result = json.load(jf)
17. arr\_std = np.std(result, ddof=1)
18. # arr\_var = np.var(result)
19. **return** arr\_std
21. **if** \_\_name\_\_ == '\_\_main\_\_':
22. averages = []
23. standards = []
24. **for** educate **in** educates:
25. average1 = average(educate)
26. averages.append(average1)
27. **for** educate **in** educates:
28. standard1 = standard(educate)
29. standards.append(standard1)
30. fig = plt.figure()
31. ax1 = fig.add\_subplot(111)
32. # axl.plot(educates,standards,"or-",label=u"标准差")
33. plt.rcParams['font.sans-serif'] = ['SimHei']
34. ax1.bar(x=educates,height=averages,width=0.7,label="平均薪资")
35. ax2 = ax1.twinx()
36. ax2.plot(educates,standards,color="y",label="标准差")
37. plt.legend()
38. plt.show()

1：调用pandas 提供的求平均值和方差的公式，求出各个学历对应水平的平均值和方差，并保存到一个列表里返回

2：利用pyplot制图，横坐标为educates ,纵坐标为薪资和标准差。在这里我们对标准差做了标识。

****



总结

从上一部分最后的图片展示可以看出，学历为专科，本科，硕士的薪资是逐级递增的，薪资与学历形成了较为明显的正相关，虽然我们的数据处理中没有涉及相关性，但是这个结果是非常明显，不言而喻的，“付出与收获成正比”，“天道酬勤”这样的话语似乎是可以用被我们的分析结果支持的。但是，请读者注意，即使是专科，也有高薪资的个例，即使是硕士也有低薪资的个例。学历是敲门砖，学历不能决定一切，但不可否认的是学历很大程度上限制了一个人的平台，所以不论是胸怀希望到社会闯一闯的，还是暂时没有方向的读者，本文都推荐继续深造，毕竟从长远考虑来看，学历更高，自身的筹码是更多的。