题目三：

【背景】

C公司是一家大型互联网公司，为了更好地保留和激励年轻优秀员工，在2019年初，人力资源部门对部分绩优年轻员工提供了长期激励，平均20万/人，统称为“Q计划”。

【任务】

举措运行一年后，领导层想要衡量Q计划的效果。你作为人力资源分析团队一员，需要通过对附件提供的数据进行分析：计算合理的指标，选取合适的分析方法，衡量Q计划在保留、激励上的效果。

思路与方法论

Part1. 数据清洗

1. 工号范围从1到14461.
2. assessment score数据库里，共有13650条2018年的数据、13641条2019年的数据
3. jm数据库里，10332条2018年的数据、13641条2019年的数据。
4. 这两个数据库的数据严重不齐全。
5. 整理学历数据，把“未知” “空” 等转换为“未知”，“硕士研究生”转换为“硕士”，“学士”转换为“本科”
6. 整理职级数据，把唯一一个缺失职级数据的硕士，用其他硕士的平均职级来替换
7. 整理在/离职数据，把“在职”转换成“0”。除了“0”，“1”等，其他的种类还有“离职”、“主动离职”、与“被动离职”等。经查，符合这三类离职原因的员工均没有2019年的评定、参与度、与满意度分数。他们可能在2019年初已经离职。
8. 整理性别数据。把**“空”**与“999”转换为“未知"___
9. 整理职位类型数据，把“空”转换为“未知”
10. 出生年份有一个异常值。有一位1900年出生、入职2002年的员工，100岁入职。

Part2. 分析数据

分为四个部分。

A. 探究Q计划参与者的基本情况

一共有***1106***人加入Q计划，截至数据记录时，Q计划参与者仍在职的为***1092***人，占比***8%***。而已经离职的Q计划参与者为***14***人。除去2019年初前离职的***9***人，2019年共离职***960***人。Q计划参与者占离职员工数量***1.45%***。在所有参与Q计划的员工中，***1.26%的Q计划参与者离职。***

B. 对已经离职的Q计划参与者的探究

在离职的Q计划参与者中：仅有两位员工在2019年的评分（AssessScore）呈现增长态势。剩下的已经离职的Q计划参与者中，一部分没有2019年的评价数据，一部分的AssessScore降低。除去其中一年记录数据缺失的员工，我发现，仅有3位离职的Q计划参与者（ID为1920、9340与3944）的参与度与满意度呈上升趋势。剩下的均呈现下降趋势。

C. 探究所有Q计划参与者的表现

1. 考虑到assessment与jm表格中的数据都严重不齐全。所以我建立了两个数据表：其中一个(df_qp)包含了所有Q计划参与者的评分、参与度、满意度信息。另一个(df_qp_paironly)仅包含了18年与19年数据都齐全的Q计划参与者信息，方便纵向比较Q计划参与者的表现情况。其中assessment表格里有1060位的Q计划参与者的数据是齐全的。其中jm表格里，仅有641位Q计划参与者的数据是齐全的。

1. 纵向比较结果：

在所有被记录了两年评分数据的Q计划参与者中：

290位参与Q计划员工的Assessment Score增长,占比27.36% 450位参与Q计划员工的Assessment Score未变,占比42.45% 320位参与Q计划员工的Assessment Score降低,占比30.19%

167位参与Q计划员工的Engagement Score增长,占比26.01% 206位参与Q计划员工的Engagement Score未变,占比32.09% 268位参与Q计划员工的Engagement Score降低,占比41.74%

159位参与Q计划员工的Satisfaction Score增长,占比24.77% 278位参与Q计划员工的Satisfaction Score未变,占比43.30% 204位参与Q计划员工的Satisfaction Score降低,占比31.78%

D.未参与Q计划的员工业绩表现分析：

一共有13355位员工未参与Q计划。这些员工的评分、满意度、参与度数据都被存储在了df_nqp数据库里。 结果如下：

2919位未参与Q计划员工的Assessment Score增长,占比24.74% 5638位未参与Q计划员工的Assessment Score未变,占比47.78% 3244位未参与Q计划员工的Assessment Score降低,占比27.49%

2037位未参与Q计划员工的Engagement Score增长,占比24.71% 2981位未参与Q计划员工的Engagement Score未变,占比36.16% 3226位未参与Q计划员工的Engagement Score降低,占比39.13%

1866位未参与Q计划员工的Satisfaction Score增长,占比22.63% 3617位未参与Q计划员工的Satisfaction Score未变,占比43.87% 2761位未参与Q计划员工的Satisfaction Score降低,占比33.49%

E.具体到员工背景信息的分析

Q计划参与者中，一共有617位硕士，440位本科生，49位未知学历 所有评分增长的Q计划参与者中，159位硕士，114位本科，17位未知 所有参与度增长的Q计划参与者中，84位硕士，76位本科，7位未知 所有满意度增长的Q计划参与者中，85位硕士，68位本科，6位未知

Q计划参与者中，一共有680位T类员工，36位S类员工，388位P类员工 所有评分增长的Q计划参与者中，187位T类员工，12位S类员工，90位P类员工 所有参与度增长的Q计划参与者中，107位T类员工，10位S类员工，50位P类员工 所有满意度增长的Q计划参与者中，98位T类员工，7位S类员工，54位P类员工

Q计划参与者中，43, 156, 180, 249, 361, 107, 9, 1 唯一的一位9级员工并没有任何的增长 所有评分增长的Q计划参与者中，10, 49, 43, 77, 82, 26, 3 (2-8) 所有参与度增长的Q计划参与者中，8, 34, 49, 53, 21, 2 (3-8) 所有满意度增长的Q计划参与者中，9, 35, 41, 54, 16, 4 (3-8)

Q计划参与者中，一共有329位男性员工，776位女性员工，1位性别未知 所有评分增长的Q计划参与者中，80位男性员工，209位女性员工，1位性别未知 所有参与度增长的Q计划参与者中，48位男性员工，119位女性员工 所有满意度增长的Q计划参与者中，50位男性员工，109位女性员工

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats
from scipy import stats
from pandas.core.frame import DataFrame
```

In [2]:

```
file_name_assessment = '/Users/guanjian/Documents/Coursera Data Science/Tencent PA 研究室 笔试/【附件三】数据/Assess_data.csv'
df_assessment = pd.read_csv(file_name_assessment)
#df.drop('Unnamed: 0',axis=1,inplace=True)
#df_copy = df.copy() #制作克隆副本
```

In [3]:

```
df_assessment[df_assessment['Period']==2018]
df_assessment[df_assessment['Period']==2019]
df_assessment
#工号范围从1到14461，两年来共有27291条数据
#assessment数据库里，2018年共有13650个数据
#assessment数据库里，2019年共有13641个数据
```

Out[3]:

	ID	Period	AssessScore
0	1	2018	5
1	2	2018	3
2	3	2018	4
3	4	2018	3
4	5	2018	5
...	...	...	...
27286	14457	2019	3
27287	14458	2019	4
27288	14459	2019	4
27289	14460	2019	3
27290	14461	2019	3

27291 rows × 3 columns

In [4]:

```python
file_name_jm = '/Users/guanjian/Documents/Coursera Data Science/Tencent PA研究室 笔试/【附件三】数据/JM_data.csv'
df_jm = pd.read_csv(file_name_jm)
#df_jm[df_jm['Year']==2018]
#df_jm[df_jm['Year']==2019]
df_jm

#工号范围从1到14461，两年来共有23346条数据
#jm数据库里，2018年共有10332个数据
#jm数据库里，2019年共有13014个数据
```

Out[4]:

	ID	Year	Engagement	Satisfaction
0	1	2018	8	8
1	2	2018	8	8
2	3	2018	7	7
3	4	2018	8	7
4	5	2018	7	8
...	...	...	...	...
23341	14457	2019	8	8
23342	14458	2019	7	8
23343	14459	2019	7	7
23344	14460	2019	8	8
23345	14461	2019	7	7

23346 rows × 4 columns

In [5]:

```
file_name_info = '/Users/guanjian/Documents/Coursera Data Science/Tencent PA研究室
 笔试/【附件三】数据/Staff_Info.csv'
df_info = pd.read_csv(file_name_info)
df_info
```

Out[5]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate
0	1	S	6	本科	0	1	1991-08-20
1	2	S	6	本科	0	1	1985-02-17
2	3	P	6	本科	0	1	1987-04-16
3	4	T	7	本科	0	0	1991-08-14
4	5	T	9	本科	0	0	1983-06-17
...	...	...	...	...	...	...	...
14456	14457	T	3	本科	0	0	1994-04-03
14457	14458	P	5	本科	0	1	1992-01-16
14458	14459	S	8	本科	0	1	1983-06-07
14459	14460	T	7	本科	0	0	1979-06-25
14460	14461	T	3	硕士	0	0	1995-08-06

14461 rows × 9 columns

Part1. 数据清洗

In [6]:

```
##数据清洗 1. 整理学历数据

df_info['EducationCode'].replace('硕士研究生','硕士',inplace=True)
df_info['EducationCode'].replace(' ','未知',inplace=True)
df_info['EducationCode'].replace('学士','本科',inplace=True)
df_info.groupby(['EducationCode']).count()
```

Out[6]:

	ID	JobClan	JobGradeRankNum	DimissionID	GenderCode	BirthDate	Career
EducationCode							
博士	54	54	54	54	54	54	
未知	1043	1043	1043	1043	1043	1043	1
本科	8034	8034	8034	8034	8034	8034	8
硕士	5286	5286	5286	5286	5286	5286	5
高中	44	44	44	44	44	44	

In [7]:

```
##数据清洗 2.整理职级数据,把唯一一个缺失职级数据的硕士，用其他硕士的平均职级来替换

df_shuoshi = df_info[df_info['EducationCode']=='硕士'].copy() #制作一个副本
#df_info.groupby(['EducationCode']).min()
df_shuoshi.drop([11937],inplace = True) #为了计算平均数，表中不能含有空格
df_shuoshi['JobGradeRankNum'] = df_shuoshi['JobGradeRankNum'].astype("int") #将所有职级数据从object转换为int形式
mean_jobgrade_shuoshi = df_shuoshi['JobGradeRankNum'].mean()
df_info['JobGradeRankNum'].replace(' ',int(mean_jobgrade_shuoshi),inplace=True)
df_info['JobGradeRankNum'] = df_info['JobGradeRankNum'].astype("int")
df_info.groupby(['JobGradeRankNum']).count()
```

Out[7]:

	ID	JobClan	EducationCode	DimissionID	GenderCode	BirthDate	CareerI
JobGradeRankNum							
1	3	3	3	3	3	3	
2	237	237	237	237	237	237	
3	792	792	792	792	792	792	
4	721	721	721	721	721	721	
5	1247	1247	1247	1247	1247	1247	1
6	3557	3557	3557	3557	3557	3557	3
7	3821	3821	3821	3821	3821	3821	3
8	2326	2326	2326	2326	2326	2326	2
9	1426	1426	1426	1426	1426	1426	1
10	259	259	259	259	259	259	
11	62	62	62	62	62	62	
12	7	7	7	7	7	7	
13	2	2	2	2	2	2	
14	1	1	1	1	1	1	

In [8]:

```
##数据清洗 3.整理离职数据
df_info['DimissionID'].replace('在职','0',inplace=True) #将“在职”替换成“0”
df_info.groupby(['DimissionID']).count()
```

Out[8]:

	ID	JobClan	JobGradeRankNum	EducationCode	GenderCode	BirthDate	Careei
DimissionID							
0	13492	13492	13492	13492	13492	13492	1
1	960	960	960	960	960	960	
主动离职	2	2	2	2	2	2	
离职	5	5	5	5	5	5	
被动离职	2	2	2	2	2	2	

In [9]:

```
df_info[(df_info['DimissionID']=='离职')|(df_info['DimissionID']=='主动离职')|(df_info['DimissionID']=='被动离职')]
```

Out[9]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate	(
146	147	P	7	硕士	离职	1	1986-03-18	:
445	446	P	4	本科	离职	0	1992-01-17	:
517	518	P	7	本科	离职	1	1982-02-11	:
799	800	P	7	未知	离职	0	1985-05-21	:
1125	1126	P	7	本科	离职	0	1985-05-18	:
1213	1214	T	7	未知	主动离职	0	1990-01-02	:
1322	1323	P	7	硕士	主动离职	0	1985-12-31	:
1377	1378	P	6	本科	被动离职	1	1990-11-14	:
1421	1422	T	8	硕士	被动离职	0	1986-07-11	:

In [10]:

```
##数据清洗 4. 清洗性别数据
df_info['GenderCode'].replace(' ','2',inplace=True)
df_info['GenderCode'].replace('999','2',inplace=True) #将所有“999”与“空格”替换成2
df_info['GenderCode'] = df_info['GenderCode'].astype('int')
df_info.groupby(['GenderCode']).count()
```

Out[10]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	BirthDate	Careei
GenderCode							
0	10191	10191	10191	10191	10191	10191	1
1	4264	4264	4264	4264	4264	4264	
2	6	6	6	6	6	6	

In [11]:

```
##数据清洗 5. 清洗职位类型特征
df_info['JobClan'].replace(' ','未知', inplace=True)
df_info.groupby(['JobClan']).count()
```

Out[11]:

	ID	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate	CareerI
JobClan							
P	5125	5125	5125	5125	5125	5125	5
S	1020	1020	1020	1020	1020	1020	1
T	8291	8291	8291	8291	8291	8291	8
未知	25	25	25	25	25	25	

In [12]:

```
## 数据清洗 6. 清洗Q计划参与与否数据
df_info.groupby(['LTI']).count()
```

Out[12]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate	C
LTI								
0	13355	13355	13355	13355	13355	13355	13355	
1	1106	1106	1106	1106	1106	1106	1106	

Part2. 分析数据

1. 基本推论

A.仍在职的Q计划参与者

In [13]:

```
df_info[(df_info['DimissionID'] == '0') & (df_info['LTI'] == 1)]
```

Out[13]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate
9	10	P	6	本科	0	1	1990-12-01
16	17	T	7	硕士	0	0	1988-06-21
24	25	P	5	本科	0	0	1992-06-13
39	40	T	4	硕士	0	1	1993-06-29
40	41	T	3	硕士	0	0	1989-08-27
...	...	...	...	...	...	...	...
14374	14375	T	6	硕士	0	0	1989-11-13
14375	14376	T	4	硕士	0	0	1993-07-28
14413	14414	P	7	本科	0	1	1993-03-14
14420	14421	T	5	本科	0	0	1992-12-25
14443	14444	T	4	硕士	0	0	1990-04-21

1092 rows × 9 columns

一共有***1106***人加入Q计划，截至数据记录时，Q计划参与者仍在职的为***1092***人，占比***8%***，离职的Q计划参与者为***14***人。除去2019年初前离职的***9***人，2019年共离职***960***人。Q计划参与者占离职员工数量***1.45%*****。**在所有参与Q计划的员工中，***1.26%的Q计划参与者离职。***

B.离职的Q计划员工，基本情况与数据参考

In [14]:

```
# 1.先看14位离职员工
df_Q_lizhi = df_info[(df_info['DimissionID'] == '1') & (df_info['LTI'] == 1)]
qlizhi_ases_list=[]
for i in range(0,14):
    period_lizhi = df_assessment[df_assessment['ID'] == df_Q_lizhi['ID'].values[i]]['Period'].values[0]
    id_qlizhi = df_Q_lizhi['ID'].values[i]
    ases_lizhi = df_assessment[df_assessment['ID'] == df_Q_lizhi['ID'].values[i]]['AssessScore'].values[0]
    pair_ingrp = [id_qlizhi,period_lizhi,ases_lizhi]
    qlizhi_ases_list.append(pair_ingrp)

    if df_assessment[df_assessment['ID'] == df_Q_lizhi['ID'].values[i]].shape[0]>1:
        period_lizhi_2 = 2019
        ases_lizhi_2 = df_assessment[df_assessment['ID'] == df_Q_lizhi['ID'].values[i]]['AssessScore'].values[1]
        pair_ingrp_2 = [id_qlizhi,period_lizhi_2,ases_lizhi_2]
        qlizhi_ases_list.append(pair_ingrp_2)

df_qlizhi_ases = DataFrame(qlizhi_ases_list)
df_qlizhi_ases.rename(columns={0:'Q计划离职员工ID',1:'年份',2:'Assess评分'}, inplace=True)
df_qlizhi_ases
```

Out[14]:

	Q计划离职员工ID	年份	Assess评分
0	659	2018	3
1	1826	2018	4
2	1826	2019	4
3	1920	2018	4
4	3189	2018	4
5	3189	2019	5
6	3944	2018	4
7	3944	2019	4
8	4976	2018	4
9	4976	2019	5
10	6459	2018	4
11	6459	2019	3
12	7663	2018	4
13	7663	2019	5
14	9340	2018	4
15	10289	2018	3
16	10656	2018	4
17	11620	2018	4
18	11620	2019	3
19	12026	2018	3
20	13291	2018	4

AssessScore一共5分。由此可见，所有离职了的14位Q计划参与者中，仅有两位员工的评分增长。剩下的，要么出现assess_score倒退，要么2019年没有评价数据。

In [15]:

```
qlizhi_egsa_list=[]
for i in range(0,14):
    period_lizhi = df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]]['Year'].values[0]
    id_qlizhi = df_Q_lizhi['ID'].values[i]
    eg_lizhi = df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]]['Engagement'].values[0]
    sa_lizhi = df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]]['Satisfaction'].values[0]

    pair_ingrp = [id_qlizhi,period_lizhi,eg_lizhi,sa_lizhi]
    qlizhi_egsa_list.append(pair_ingrp)

    if df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]].shape[0]>1:
        period_lizhi_2 = 2019
        eg_lizhi_2 = df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]]['Engagement'].values[1]
        sa_lizhi_2 = df_jm[df_jm['ID'] == df_Q_lizhi['ID'].values[i]]['Satisfaction'].values[1]

        pair_ingrp_2 = [id_qlizhi,period_lizhi_2,eg_lizhi_2,sa_lizhi_2]
        qlizhi_egsa_list.append(pair_ingrp_2)

df_qlizhi_egsa = DataFrame(qlizhi_egsa_list)
df_qlizhi_egsa.rename(columns={0:'Q计划离职员工ID',1:'年份',2:'Engagement评分',3:'满意度评分'}, inplace=True)
df_qlizhi_egsa
```

Out[15]:

	Q计划离职员工ID	年份	Engagement评分	满意度评分
0	659	2018	9	9
1	659	2019	8	8
2	1826	2019	4	4
3	1920	2018	5	6
4	1920	2019	10	9
5	3189	2018	10	9
6	3189	2019	8	8
7	3944	2018	8	8
8	3944	2019	8	8
9	4976	2018	5	6
10	6459	2019	8	9
11	7663	2018	8	9
12	7663	2019	6	7
13	9340	2018	8	9
14	9340	2019	10	10
15	10289	2018	10	10
16	10289	2019	9	9
17	10656	2019	6	7
18	11620	2018	7	7
19	11620	2019	6	7
20	12026	2019	4	5
21	13291	2018	8	8

Engagement与Satisfaction的满分均为10分。除去其中一年记录数据缺失的员工，我们可以发现，仅有ID为1920、9340与3944的离职员工的参与度与满意度呈上升趋势。

C.整体Q计划员工分析

In [16]:

```
#第一步，选出所有参与Q计划的员工id列表，存入QP_idlist
df_QP = df_info[(df_info['LTI'] == 1)].copy()
QP_idlist = df_QP['ID'].values
QP_idlist #此list里存储了所有参与Q计划的员工ID
```

Out[16]:

```
array([   10,    17,    25, ..., 14414, 14421, 14444])
```

In [17]:

```
#建立总的大表存储所有Q计划员工的评价信息
df_qp = pd.DataFrame(columns=['id1','year1','assessment','fenge','id2','year2','engagement','satisfaction'])
```

In [18]:

```
##往大表里添加进assessment数据
i = 0
df_qp_index_1 = 0

while i<1106:

    period_qp = df_assessment[df_assessment['ID'] == QP_idlist[i]]['Period'].values[0]
    id_qp = QP_idlist[i]
    ases_qp = df_assessment[df_assessment['ID'] == QP_idlist[i]]['AssessScore'].values[0]

    df_qp.loc[df_qp_index_1,'id1'] = id_qp
    df_qp.loc[df_qp_index_1,'year1'] = period_qp
    df_qp.loc[df_qp_index_1,'assessment'] = ases_qp

    if df_assessment[df_assessment['ID'] == QP_idlist[i]].shape[0]>1:

        period_qp_2 = 2019
        ases_qp_2 = df_assessment[df_assessment['ID'] == QP_idlist[i]]['AssessScore'].values[1]

        df_qp.loc[df_qp_index_1+1,'id1'] = id_qp
        df_qp.loc[df_qp_index_1+1,'year1'] = period_qp_2
        df_qp.loc[df_qp_index_1+1,'assessment'] = ases_qp_2

        df_qp_index_1+=1

    df_qp_index_1+=1

    i+=1
```

In [19]:

```
k=0
df_qp_index_2 = 0

while k<1106:

    period_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Year'].values[0]
    id_qp = QP_idlist[k]
    eg_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Engagement'].values[0]
    sa_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Satisfaction'].values[0]

    df_qp.loc[df_qp_index_2,'id2'] = id_qp
    df_qp.loc[df_qp_index_2,'year2'] = period_qp
    df_qp.loc[df_qp_index_2,'engagement'] = eg_qp
    df_qp.loc[df_qp_index_2,'satisfaction'] = sa_qp


    if df_jm[df_jm['ID'] == QP_idlist[k]].shape[0]>1:

        period_qp_2 = 2019
        eg_qp_2 = df_jm[df_jm['ID'] == QP_idlist[k]]['Engagement'].values[1]
        sa_qp_2 = df_jm[df_jm['ID'] == QP_idlist[k]]['Satisfaction'].values[1]

        df_qp.loc[df_qp_index_2+1,'id2'] = id_qp
        df_qp.loc[df_qp_index_2+1,'year2'] = period_qp_2
        df_qp.loc[df_qp_index_2+1,'engagement'] = eg_qp_2
        df_qp.loc[df_qp_index_2+1,'satisfaction'] = sa_qp_2

        df_qp_index_2+=1

    df_qp_index_2+=1
    k+=1
```

In [20]:

```
df_qp['fenge'].replace(np.nan,' ',inplace=True)
df_qp
```

Out[20]:

	id1	year1	assessment	fenge	id2	year2	engagement	satisfaction
0	10	2018	3		10	2018	4	5
1	10	2019	3		10	2019	5	5
2	17	2018	3		17	2018	10	9
3	17	2019	3		17	2019	10	9
4	25	2018	4		25	2018	7	8
...	...	...	...	...	...	...	...	...
2161	14414	2019	4		NaN	NaN	NaN	NaN
2162	14421	2018	3		NaN	NaN	NaN	NaN
2163	14421	2019	3		NaN	NaN	NaN	NaN
2164	14444	2018	4		NaN	NaN	NaN	NaN
2165	14444	2019	3		NaN	NaN	NaN	NaN

2166 rows × 8 columns

In [21]:

```
##判断在df_assessment中有多少个数据点仅仅有一年的数据
df_qp.groupby(['id1']).count()
counter = 0
for k in range(0,1106):
    if df_assessment[df_assessment['ID'] == QP_idlist[k]].shape[0]==1:
        counter+=1
print(counter)
```

```
46
```

In [22]:

```
##判断在df_jm中有多少个数据点仅仅有一年的数据
df_qp.groupby(['id2']).count()
counter = 0
for k in range(0,1106):
    if df_jm[df_jm['ID'] == QP_idlist[k]].shape[0]==1:
        counter+=1
print(counter)
```

```
465
```

In [23]:

```
#判断数据点有没有不在评价分数数据库里的。结果显示：没有。Q计划员工都有评分
for k in range(0,1106):
    if QP_idlist[k] not in df_jm['ID'].values:
        print('呀！')
```

In [24]:

```
df_qp.groupby(['id1']).count()['id2'].sum()
```

Out[24]:

```
1747
```

df_qp存储了所有的数据，用该表里2019年的数据去横向比较非Q计划的员工。

df_qp_paironly存储了所有Q计划员工中拥有成对（两年）数据的数据。方便进行纵向比较。比较该员工在参与Q计划前后的业绩增长或后退。

In [25]:

```
df_qp_paironly = pd.DataFrame(columns=['id1','year1','assessment','fenge','id2','year2','engagement','satisfaction'])
```

In [26]:

```
##往大表里添加进assessment数据
i = 0
df_qp_index_3 = 0

while i<1106:

    period_qp = df_assessment[df_assessment['ID'] == QP_idlist[i]]['Period'].values
[0]
    id_qp = QP_idlist[i]
    ases_qp = df_assessment[df_assessment['ID'] == QP_idlist[i]]['AssessScore'].val
ues[0]

    df_qp_paironly.loc[df_qp_index_3,'id1'] = id_qp
    df_qp_paironly.loc[df_qp_index_3,'year1'] = period_qp
    df_qp_paironly.loc[df_qp_index_3,'assessment'] = ases_qp

    if df_assessment[df_assessment['ID'] == QP_idlist[i]].shape[0]>1:

        period_qp_2 = 2019
        ases_qp_2 = df_assessment[df_assessment['ID'] == QP_idlist[i]]['AssessScor
e'].values[1]

        df_qp_paironly.loc[df_qp_index_3+1,'id1'] = id_qp
        df_qp_paironly.loc[df_qp_index_3+1,'year1'] = period_qp_2
        df_qp_paironly.loc[df_qp_index_3+1,'assessment'] = ases_qp_2

        df_qp_index_3 += 2

    #df_qp_index_1+=1

    i+=1
```

In [27]:

```
k=0
df_qp_index_4 = 0

while k<1106:

    period_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Year'].values[0]
    id_qp = QP_idlist[k]
    eg_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Engagement'].values[0]
    sa_qp = df_jm[df_jm['ID'] == QP_idlist[k]]['Satisfaction'].values[0]

    df_qp_paironly.loc[df_qp_index_4,'id2'] = id_qp
    df_qp_paironly.loc[df_qp_index_4,'year2'] = period_qp
    df_qp_paironly.loc[df_qp_index_4,'engagement'] = eg_qp
    df_qp_paironly.loc[df_qp_index_4,'satisfaction'] = sa_qp


    if df_jm[df_jm['ID'] == QP_idlist[k]].shape[0]>1:

        period_qp_2 = 2019
        eg_qp_2 = df_jm[df_jm['ID'] == QP_idlist[k]]['Engagement'].values[1]
        sa_qp_2 = df_jm[df_jm['ID'] == QP_idlist[k]]['Satisfaction'].values[1]

        df_qp_paironly.loc[df_qp_index_4+1,'id2'] = id_qp
        df_qp_paironly.loc[df_qp_index_4+1,'year2'] = period_qp_2
        df_qp_paironly.loc[df_qp_index_4+1,'engagement'] = eg_qp_2
        df_qp_paironly.loc[df_qp_index_4+1,'satisfaction'] = sa_qp_2

        df_qp_index_4+=2

    #df_qp_index_2+=1
    k+=1
```

In [28]:

```
df_qp_paironly
```

Out[28]:

	id1	year1	assessment	fenge	id2	year2	engagement	satisfaction
0	10	2018	3	NaN	10	2018	4	5
1	10	2019	3	NaN	10	2019	5	5
2	17	2018	3	NaN	17	2018	10	9
3	17	2019	3	NaN	17	2019	10	9
4	25	2018	4	NaN	25	2018	7	8
...	...	...	...	...	...	...	...	...
2115	14414	2019	4	NaN	NaN	NaN	NaN	NaN
2116	14421	2018	3	NaN	NaN	NaN	NaN	NaN
2117	14421	2019	3	NaN	NaN	NaN	NaN	NaN
2118	14444	2018	4	NaN	NaN	NaN	NaN	NaN
2119	14444	2019	3	NaN	NaN	NaN	NaN	NaN

2120 rows × 8 columns

In [29]:

```
df_qp_paironly.groupby(['id1']).count()['id2'].sum()
```

Out[29]:

```
1282
```

In [30]:

```
list_qp_id_paironly_ases = df_qp_paironly['id1'].unique()
list_qp_id_paironly_ases.size
```

Out[30]:

```
1060
```

In [31]:

```
counter_ases_posi = 0
counter_ases_eqal = 0
counter_ases_nega = 0

list_id_ases_posi = []
for i in range(0,list_qp_id_paironly_ases.size):
    df_demo = df_qp_paironly[df_qp_paironly['id1']==list_qp_id_paironly_ases[i]]
    index=2*(i+1)-1
    diff = df_demo['assessment'][index]-df_demo['assessment'][index-1]
    #print(i)
    if diff > 0:
        list_id_ases_posi.append(list_qp_id_paironly_ases[i])
        counter_ases_posi += 1
    elif diff == 0:
        counter_ases_eqal += 1
    else:
        counter_ases_nega += 1

prct_ases_posi = counter_ases_posi/list_qp_id_paironly_ases.size
prct_ases_eqal = counter_ases_eqal/list_qp_id_paironly_ases.size
prct_ases_nega = counter_ases_nega/list_qp_id_paironly_ases.size

print(str(counter_ases_posi)+'位参与Q计划员工的Assessment Score增长,占比'+'{:.2%}'.form
at(prct_ases_posi))
print(str(counter_ases_eqal)+'位参与Q计划员工的Assessment Score未变,占比'+'{:.2%}'.form
at(prct_ases_eqal))
print(str(counter_ases_nega)+'位参与Q计划员工的Assessment Score降低,占比'+'{:.2%}'.form
at(prct_ases_nega))
```

```
290位参与Q计划员工的Assessment Score增长,占比27.36%
450位参与Q计划员工的Assessment Score未变,占比42.45%
320位参与Q计划员工的Assessment Score降低,占比30.19%
```

In [32]:

```
list_qp_id_paironly_egsa = df_qp_paironly['id2'].unique()
list_qp_id_paironly_egsa.size #该list里最后一位存储了NaN, 因为不同于assessment score, eg
sa的表在评分总表里的最后全部是以nan占的位
```

Out[32]:

```
642
```

In [33]:

```
counter_eg_posi = 0
counter_eg_eqal = 0
counter_eg_nega = 0

counter_sa_posi = 0
counter_sa_eqal = 0
counter_sa_nega = 0

list_id_sa_posi = [] # 存储有satisfaction分数增高的Q计划员工ID
list_id_eg_posi = [] # 存储有engagement分数增高的Q计划员工ID
for i in range(0,list_qp_id_paironly_egsa.size-1):
    df_demo = df_qp_paironly[df_qp_paironly['id2']==list_qp_id_paironly_egsa[i]]
    index=2*(i+1)-1
    try:
        diff_eg = df_demo['engagement'][index]-df_demo['engagement'][index-1]
    except:
        print('出问题的是'+str(i))
    diff_sa = df_demo['satisfaction'][index]-df_demo['satisfaction'][index-1]

    #print(i)
    if diff_eg > 0:
        list_id_eg_posi.append(list_qp_id_paironly_egsa[i])
        counter_eg_posi += 1
    elif diff_eg == 0:
        counter_eg_eqal += 1
    else:
        counter_eg_nega += 1

    if diff_sa > 0:
        list_id_sa_posi.append(list_qp_id_paironly_egsa[i])
        counter_sa_posi += 1
    elif diff_sa == 0:
        counter_sa_eqal += 1
    else:
        counter_sa_nega += 1


prct_eg_posi = counter_eg_posi/list_qp_id_paironly_egsa.size
prct_eg_eqal = counter_eg_eqal/list_qp_id_paironly_egsa.size
prct_eg_nega = counter_eg_nega/list_qp_id_paironly_egsa.size

prct_sa_posi = counter_sa_posi/list_qp_id_paironly_egsa.size
prct_sa_eqal = counter_sa_eqal/list_qp_id_paironly_egsa.size
prct_sa_nega = counter_sa_nega/list_qp_id_paironly_egsa.size

print(str(counter_eg_posi)+'位参与Q计划员工的Engagement Score增长,占比'+'{:.2%}'.format
(prct_eg_posi))
print(str(counter_eg_eqal)+'位参与Q计划员工的Engagement Score未变,占比'+'{:.2%}'.format
(prct_eg_eqal))
print(str(counter_eg_nega)+'位参与Q计划员工的Engagement Score降低,占比'+'{:.2%}'.format
(prct_eg_nega))
print('--')
print(str(counter_sa_posi)+'位参与Q计划员工的Satisfaction Score增长,占比'+'{:.2%}'.form
at(prct_sa_posi))
print(str(counter_sa_eqal)+'位参与Q计划员工的Satisfaction Score未变,占比'+'{:.2%}'.form
at(prct_sa_eqal))
print(str(counter_sa_nega)+'位参与Q计划员工的Satisfaction Score降低,占比'+'{:.2%}'.form
at(prct_sa_nega))
```

```
167位参与Q计划员工的Engagement Score增长,占比26.01%
206位参与Q计划员工的Engagement Score未变,占比32.09%
268位参与Q计划员工的Engagement Score降低,占比41.74%
--
159位参与Q计划员工的Satisfaction Score增长,占比24.77%
278位参与Q计划员工的Satisfaction Score未变,占比43.30%
204位参与Q计划员工的Satisfaction Score降低,占比31.78%
```

D.整体非Q计划员工与Q计划员工的比较

In [34]:

```
#第一步，选出所有未参与Q计划的员工id列表，存入NQP_idlist
df_nqp = df_info[(df_info['LTI'] == 0)].copy()
nqp_idlist = df_nqp['ID'].values
nqp_idlist.size #此list里存储了所有参与Q计划的员工ID
```

Out[34]:

```
13355
```

In [35]:

```
#建立总的大表存储所有未参与Q计划员工的评价信息
df_nqp = pd.DataFrame(columns=['id1','year1','assessment','fenge','id2','year2','en
gagement','satisfaction'])
df_nqp_ases = pd.DataFrame(columns=['id','year','assessment'])
```

In [36]:

```
df_nqp_ases=df_assessment[~df_assessment['ID'].isin(QP_idlist)]
df_nqp_ases
```

Out[36]:

	ID	Period	AssessScore
0	1	2018	5
1	2	2018	3
2	3	2018	4
3	4	2018	3
4	5	2018	5
...	...	...	...
27286	14457	2019	3
27287	14458	2019	4
27288	14459	2019	4
27289	14460	2019	3
27290	14461	2019	3

25125 rows × 3 columns

In [37]:

```
counter_nqp_ases_posi = 0
counter_nqp_ases_eqal = 0
counter_nqp_ases_nega = 0
counter_effective_times = 0
list_nqp_id_ases_posi = []
for i in range(0,nqp_idlist.size):

    df_demo = df_nqp_ases[df_nqp_ases['ID']==nqp_idlist[i]].copy()
    if df_demo.shape[0]>1:
        df_demo.index=[0,1]
        diff = df_demo['AssessScore'][1]-df_demo['AssessScore'][0]

        if diff > 0:
            list_nqp_id_ases_posi.append(nqp_idlist[i])
            counter_nqp_ases_posi += 1
        elif diff == 0:
            counter_nqp_ases_eqal += 1
        else:
            counter_nqp_ases_nega += 1
        counter_effective_times+=1

prct_nqp_ases_posi = counter_nqp_ases_posi / counter_effective_times #nqp_idlist.size
prct_nqp_ases_eqal = counter_nqp_ases_eqal / counter_effective_times #nqp_idlist.size
prct_nqp_ases_nega = counter_nqp_ases_nega / counter_effective_times #nqp_idlist.size

print(str(counter_nqp_ases_posi)+'位未参与Q计划员工的Assessment Score增长,占比'+'{:.2%}'.format(prct_nqp_ases_posi))
print(str(counter_nqp_ases_eqal)+'位未参与Q计划员工的Assessment Score未变,占比'+'{:.2%}'.format(prct_nqp_ases_eqal))
print(str(counter_nqp_ases_nega)+'位未参与Q计划员工的Assessment Score降低,占比'+'{:.2%}'.format(prct_nqp_ases_nega))
```

```
2919位未参与Q计划员工的Assessment Score增长,占比24.74%
5638位未参与Q计划员工的Assessment Score未变,占比47.78%
3244位未参与Q计划员工的Assessment Score降低,占比27.49%
```

In [38]:

```
df_nqp_egsa=df_jm[~df_jm['ID'].isin(QP_idlist)]
df_nqp_egsa
```

Out[38]:

	ID	Year	Engagement	Satisfaction
0	1	2018	8	8
1	2	2018	8	8
2	3	2018	7	7
3	4	2018	8	7
4	5	2018	7	8
...	...	...	...	...
23341	14457	2019	8	8
23342	14458	2019	7	8
23343	14459	2019	7	7
23344	14460	2019	8	8
23345	14461	2019	7	7

21599 rows × 4 columns

In [39]:

```
counter_nqp_eg_posi = 0
counter_nqp_eg_eqal = 0
counter_nqp_eg_nega = 0

counter_nqp_sa_posi = 0
counter_nqp_sa_eqal = 0
counter_nqp_sa_nega = 0
counter_effective_times_1 = 0

list_nqpid_sa_posi = [] # 存储有satisfaction分数增高非Q计划员工ID
list_nqpid_eg_posi = [] # 存储有engagement分数增高的非Q计划员工ID

for i in range(0,nqp_idlist.size):

    df_demo = df_nqp_egsa[df_nqp_egsa['ID']==nqp_idlist[i]].copy()
    if df_demo.shape[0]>1:
        df_demo.index=[0,1]

        diff_eg = df_demo['Engagement'][1]-df_demo['Engagement'][0]
        diff_sa = df_demo['Satisfaction'][1]-df_demo['Satisfaction'][0]

        if diff_eg > 0:
            list_nqpid_eg_posi.append(nqp_idlist[i])
            counter_nqp_eg_posi += 1
        elif diff_eg == 0:
            counter_nqp_eg_eqal += 1
        else:
            counter_nqp_eg_nega += 1

        if diff_sa > 0:
            list_nqpid_sa_posi.append(nqp_idlist[i])
            counter_nqp_sa_posi += 1
        elif diff_sa == 0:
            counter_nqp_sa_eqal += 1
        else:
            counter_nqp_sa_nega += 1

        counter_effective_times_1 += 1

prct_nqp_eg_posi = counter_nqp_eg_posi / counter_effective_times_1
prct_nqp_eg_eqal = counter_nqp_eg_eqal / counter_effective_times_1
prct_nqp_eg_nega = counter_nqp_eg_nega / counter_effective_times_1

prct_nqp_sa_posi = counter_nqp_sa_posi / counter_effective_times_1
prct_nqp_sa_eqal = counter_nqp_sa_eqal / counter_effective_times_1
prct_nqp_sa_nega = counter_nqp_sa_nega / counter_effective_times_1

print(str(counter_nqp_eg_posi)+'位未参与Q计划员工的Engagement Score增长,占比'+'{:.2%}'.
format(prct_nqp_eg_posi))
print(str(counter_nqp_eg_eqal)+'位未参与Q计划员工的Engagement Score未变,占比'+'{:.2%}'.
format(prct_nqp_eg_eqal))
print(str(counter_nqp_eg_nega)+'位未参与Q计划员工的Engagement Score降低,占比'+'{:.2%}'.
format(prct_nqp_eg_nega))
print('--')
print(str(counter_nqp_sa_posi)+'位未参与Q计划员工的Satisfaction Score增长,占比'+'{:.2%}
'.format(prct_nqp_sa_posi))
print(str(counter_nqp_sa_eqal)+'位未参与Q计划员工的Satisfaction Score未变,占比'+'{:.2%}
'.format(prct_nqp_sa_eqal))
print(str(counter_nqp_sa_nega)+'位未参与Q计划员工的Satisfaction Score降低,占比'+'{:.2%}
'.format(prct_nqp_sa_nega))
```

```
2037位未参与Q计划员工的Engagement Score增长,占比24.71%
2981位未参与Q计划员工的Engagement Score未变,占比36.16%
3226位未参与Q计划员工的Engagement Score降低,占比39.13%
--
1866位未参与Q计划员工的Satisfaction Score增长,占比22.63%
3617位未参与Q计划员工的Satisfaction Score未变,占比43.87%
2761位未参与Q计划员工的Satisfaction Score降低,占比33.49%
```

E. Q计划员工背景信息分析

In [40]:

```
list_id_ases_posi
list_id_eg_posi
list_id_sa_posi
##总体Q计划参与员工（无论离职与否）信息表格
df_info[df_info['ID'].isin(QP_idlist)]
```

Out[40]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate
9	10	P	6	本科	0	1	1990-12-01
16	17	T	7	硕士	0	0	1988-06-21
24	25	P	5	本科	0	0	1992-06-13
39	40	T	4	硕士	0	1	1993-06-29
40	41	T	3	硕士	0	0	1989-08-27
...	...	...	...	...	...	...	...
14374	14375	T	6	硕士	0	0	1989-11-13
14375	14376	T	4	硕士	0	0	1993-07-28
14413	14414	P	7	本科	0	1	1993-03-14
14420	14421	T	5	本科	0	0	1992-12-25
14443	14444	T	4	硕士	0	0	1990-04-21

1106 rows × 9 columns

In [41]:

```
#Q计划参与者中，一共有617位硕士，440位本科生，49位未知学历
#所有评分增长的Q计划参与者中，159位硕士，114位本科，17位未知
#所有参与度增长的Q计划参与者中，84位硕士，76位本科，7位未知
#所有满意度增长的Q计划参与者中，85位硕士，68位本科，6位未知

#Q计划参与者中，一共有680位T类员工，36位S类员工，388位P类员工
#所有评分增长的Q计划参与者中，187位T类员工，12位S类员工，90位P类员工
#所有参与度增长的Q计划参与者中，107位T类员工，10位S类员工，50位P类员工
#所有满意度增长的Q计划参与者中，98位T类员工，7位S类员工，54位P类员工

#Q计划参与者中，43，156，180，249，361，107，   9，   1 唯一的一位9级员工并没有任何的增长
#所有评分增长的Q计划参与者中，10，49，43，77，82，26，  3  (2-8)
#所有参与度增长的Q计划参与者中，8，34，49，53，21，  2 (3-8)
#所有满意度增长的Q计划参与者中，9，35，41，54，16，  4 (3-8)

#Q计划参与者中，一共有329位男性员工，776位女性员工，1位性别未知
#所有评分增长的Q计划参与者中，80位男性员工，209位女性员工，1位性别未知
#所有参与度增长的Q计划参与者中，48位男性员工，119位女性员工
#所有满意度增长的Q计划参与者中，50位男性员工，109位女性员工
```

In [42]:

```
#分职位等级去看表现更加的Q计划员工背景情况
df_info[df_info['ID'].isin(list_id_ases_posi)].groupby(['JobGradeRankNum']).count()
df_info[df_info['ID'].isin(list_id_eg_posi)].groupby(['JobGradeRankNum']).count()
df_info[df_info['ID'].isin(list_id_sa_posi)].groupby(['JobGradeRankNum']).count()
```

Out[42]:

	ID	JobClan	EducationCode	DimissionID	GenderCode	BirthDate	CareerDat
JobGradeRankNum							
3	9	9	9	9	9	9	
4	35	35	35	35	35	35	3
5	41	41	41	41	41	41	4
6	54	54	54	54	54	54	5
7	16	16	16	16	16	16	1
8	4	4	4	4	4	4	

In [43]:

```
#分性别去看表现更加的Q计划员工背景情况
df_info[df_info['ID'].isin(list_id_ases_posi)].groupby(['GenderCode']).count()
df_info[df_info['ID'].isin(list_id_eg_posi)].groupby(['GenderCode']).count()
df_info[df_info['ID'].isin(list_id_sa_posi)].groupby(['GenderCode']).count()
```

Out[43]:

	ID	JobClan	JobGradeRankNum	EducationCode	DimissionID	BirthDate	CareerD
GenderCode							
0	109	109	109	109	109	109	1
1	50	50	50	50	50	50	

In [44]:

```
#分工作类别去看表现更加的Q计划员工背景情况
df_info[df_info['ID'].isin(list_id_ases_posi)].groupby(['JobClan']).count()
df_info[df_info['ID'].isin(list_id_eg_posi)].groupby(['JobClan']).count()
df_info[df_info['ID'].isin(list_id_sa_posi)].groupby(['JobClan']).count()
```

Out[44]:

	ID	JobGradeRankNum	EducationCode	DimissionID	GenderCode	BirthDate	CareerDat
JobClan							
P	54	54	54	54	54	54	5
S	7	7	7	7	7	7	
T	98	98	98	98	98	98	9

In [45]:

```
#分学历去看表现更加的Q计划员工背景情况
df_info[df_info['ID'].isin(list_id_ases_posi)].groupby(['EducationCode']).count()
df_info[df_info['ID'].isin(list_id_eg_posi)].groupby(['EducationCode']).count()
df_info[df_info['ID'].isin(list_id_sa_posi)].groupby(['EducationCode']).count()
```

Out[45]:

	ID	JobClan	JobGradeRankNum	DimissionID	GenderCode	BirthDate	CareerDat
EducationCode							
未知	6	6	6	6	6	6	
本科	68	68	68	68	68	68	6
硕士	85	85	85	85	85	85	8