Project 8 Solutions

Anuj Thakkar

Collaborators: (Collaborators listed here. Include names, which part of the project you gave or sought help with, and how you helped or were helped.)

TA help: None

Online resources used: None

Question 1

```
import pandas as pd
```

```
import pandas as pd
users = pd.read_parquet("/class/datamine/data/yelp/data/parquet/users.parquet")
reviews = pd.read_parquet("/class/datamine/data/yelp/data/parquet/reviews.parquet")
def get_friends_data(user_id: str) -> pd.DataFrame:
  myDF = pd.DataFrame()
  for i in range(0, users.shape[0]):
    if (users['user_id'][i] == user_id):
      mylist = users['friends'][i].split(", ")
  for j in range(0, users.shape[0]):
    if (users['user_id'][j] in mylist):
      myDF = myDF.append(users.iloc[j])
  return myDF

print(get_friends_data("ntlvfPzc8eglqvk92iDIAw").shape) # (13,22)
```

```
(13, 22)
```

```
print(get_friends_data("AY-laIws3S7YXNl_f_D6rQ").shape) # (1, 22)
```

```
(1, 22)
```

```
print(get_friends_data("xvu8G900tezTzbbfqmTKvA").shape) # (193,22)
```

```
(193, 22)
```

Question 2

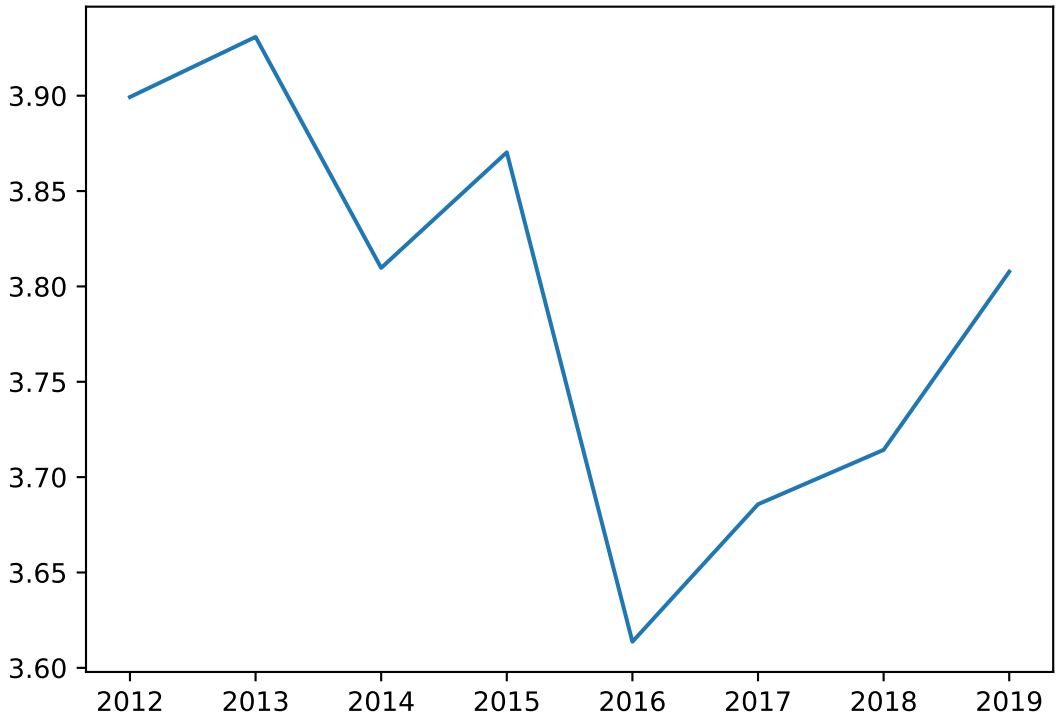
```
def calculate_avg_business_stars(business_id : str) -> float:
  temp = reviews[reviews["business_id"] == business_id]
  return sum(temp["stars"]) / len(temp["stars"])


print(calculate_avg_business_stars("f9NumwFMBDn751xgFiRbNA")) # 3.1025641025641026
```

```
3.1025641025641026
```

Question 3

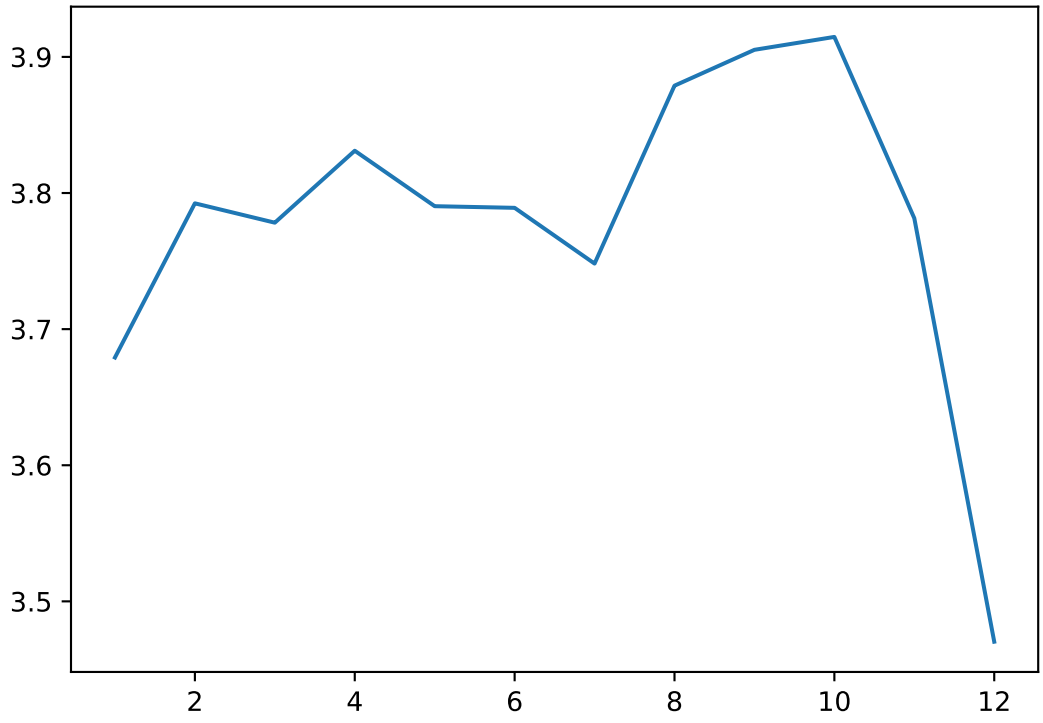
```
import matplotlib.pyplot as plt

def visualize_stars_over_time(my_business_id: str):

  myyears = []
  for i in range(0,reviews.shape[0]):
    myyears.append(reviews['date'][i].year)
  reviews['year'] = myyears
  averagestars = reviews.groupby(['business_id','year'],as_index=False)['stars'].mean()
  mydict = {}
  for i in range(0,averagestars.shape[0]):
    if averagestars['business_id'][i] == my_business_id:
      mydict[averagestars['year'][i]] = averagestars['stars'][i]
  print (mydict.keys())
  print (mydict.values())
  plt.plot(mydict.keys(), mydict.values())

  plt.show()
  plt.close()
  return None

visualize_stars_over_time('RESDUcs7fIiihp38-d6_6g')
```

```
dict_keys([2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019])
dict_values([3.899297423887588, 3.930833333333333, 3.8097043214556483,
3.8703312191684285, 3.6137026239067054, 3.6857682619647356, 3.7142857142857144,
3.807715133531157])
```



Question 4

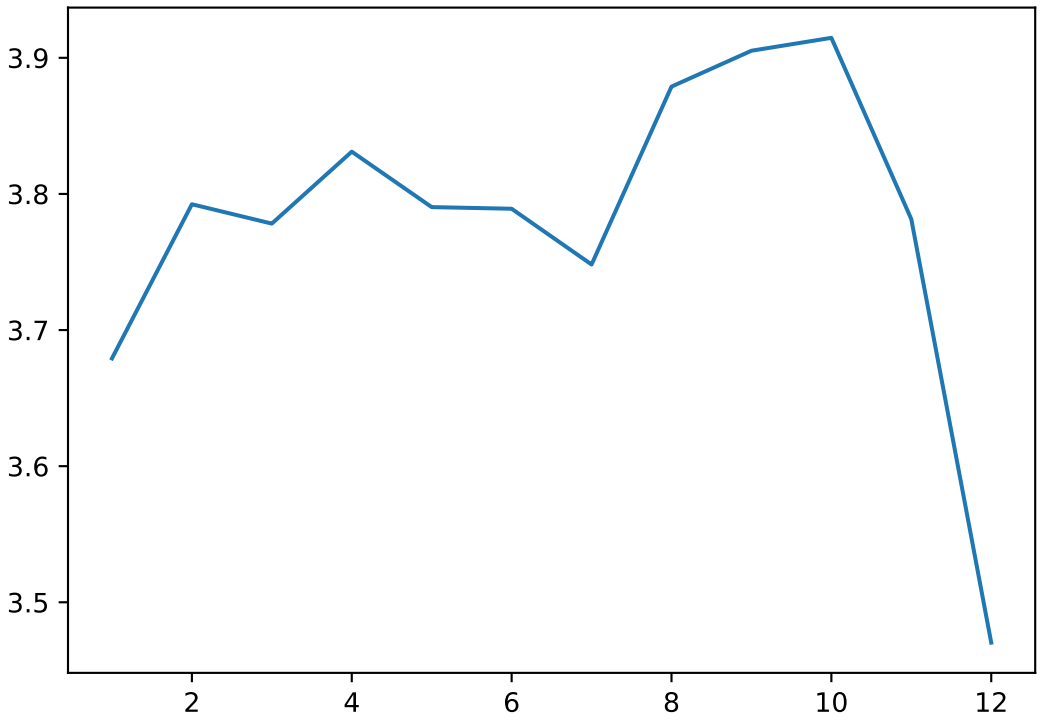
```
def visualize_stars_over_time(my_business_id: str, granularity: str = "years"):

  myyearsormonths = []
  if granularity == "months":
    for i in range(0,reviews.shape[0]):
      myyearsormonths.append(reviews['date'][i].month)
  else:
    for i in range(0, reviews.shape[0]):
      myyearsormonths.append(reviews['date'][i].year)
  reviews['yearormonth'] = myyearsormonths
  averagestars = reviews.groupby(['business_id','yearormonth'],as_index=False)['stars'].mean()
  mydict = {}
  for i in range(0,averagestars.shape[0]):
    if averagestars['business_id'][i] == my_business_id:
      mydict[averagestars['yearormonth'][i]] = averagestars['stars'][i]
  plt.plot(mydict.keys(), mydict.values())

  plt.show()
  plt.close()

visualize_stars_over_time('RESDUcs7fIiihp38-d6_6g', "months")
```

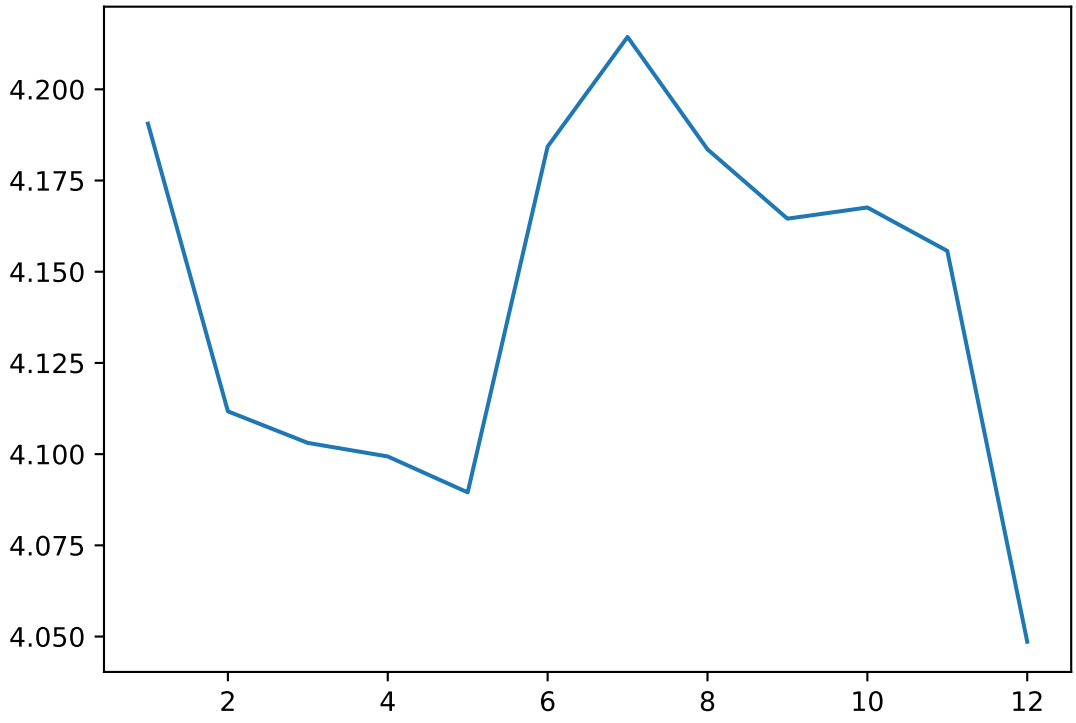


Question 5

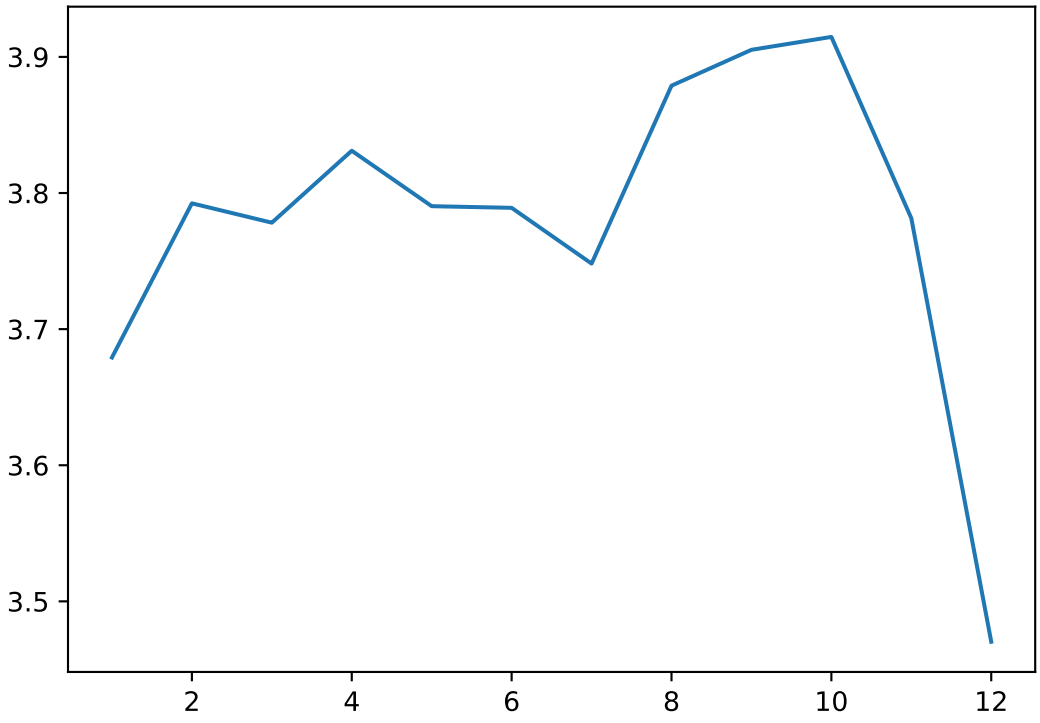
```
def visualize_stars_over_time(*args, **kwargs):

  granularity = ""
  businesses = []
  if (len(kwargs) > 0):
    granularity = kwargs["granularity"]
    businesses = args
  else:
    granularity = args[len(args)-1]
    businesses = args[0:len(args)-1]
  for my_business_id in businesses:
    myyearsormonths = []
    if granularity == "months":
      for i in range(0,reviews.shape[0]):
        myyearsormonths.append(reviews['date'][i].month)
    else:
      for i in range(0, reviews.shape[0]):
        myyearsormonths.append(reviews['date'][i].year)
    reviews['yearormonth'] = myyearsormonths
    averagestars = reviews.groupby(['business_id','yearormonth'],as_index=False)['stars'].mean()
    mydict = {}
    for i in range(0,averagestars.shape[0]):
      if averagestars['business_id'][i] == my_business_id:
```

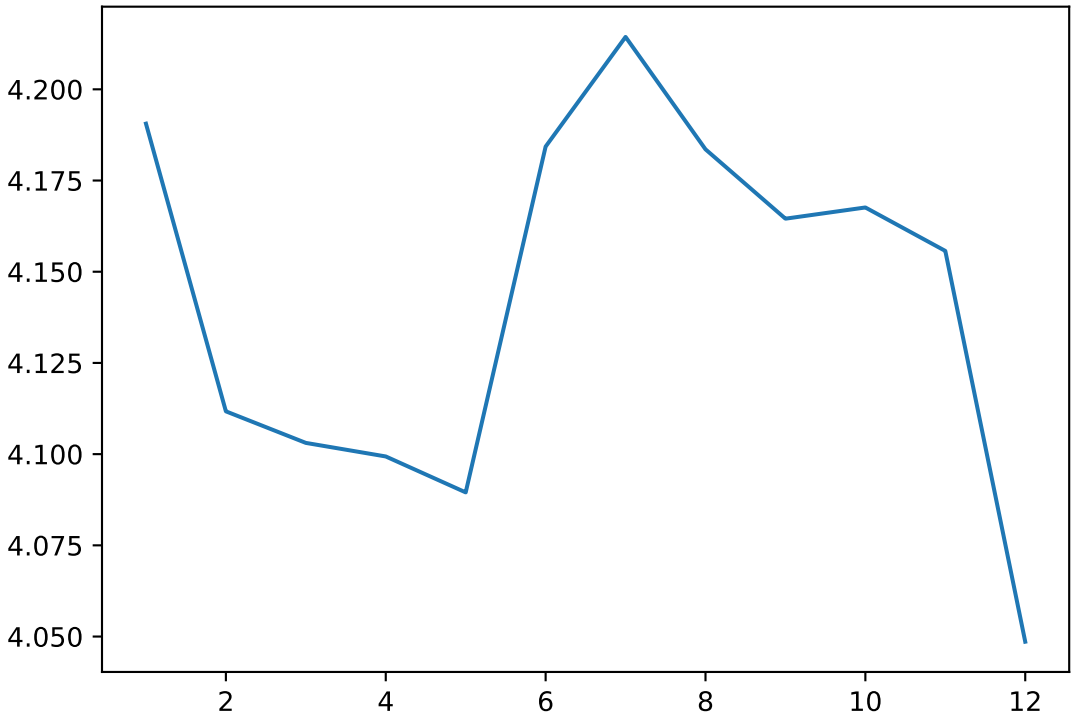
```
        mydict[averagestars['yearormonth'][i]] = averagestars['stars'][i]
    plt.plot(mydict.keys(), mydict.values())
    plt.show()
    plt.close()

visualize_stars_over_time("RESDUcs7fIiihp38-d6_6g", "4JNXUYY8wbaaDmk3BPzlWw", "months")
```

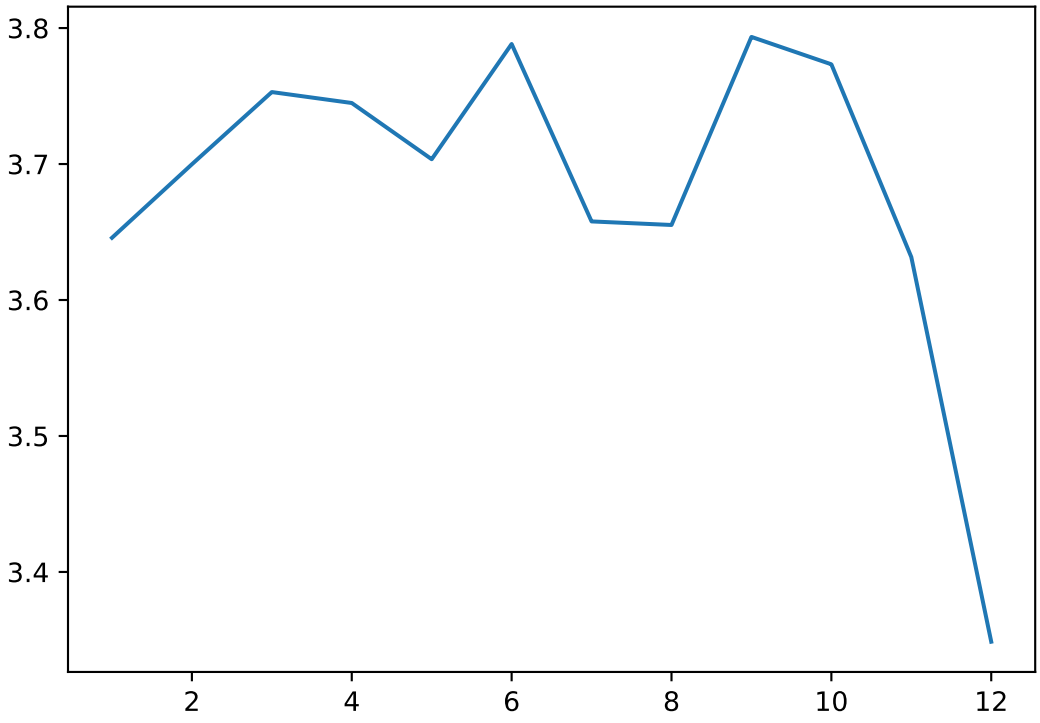




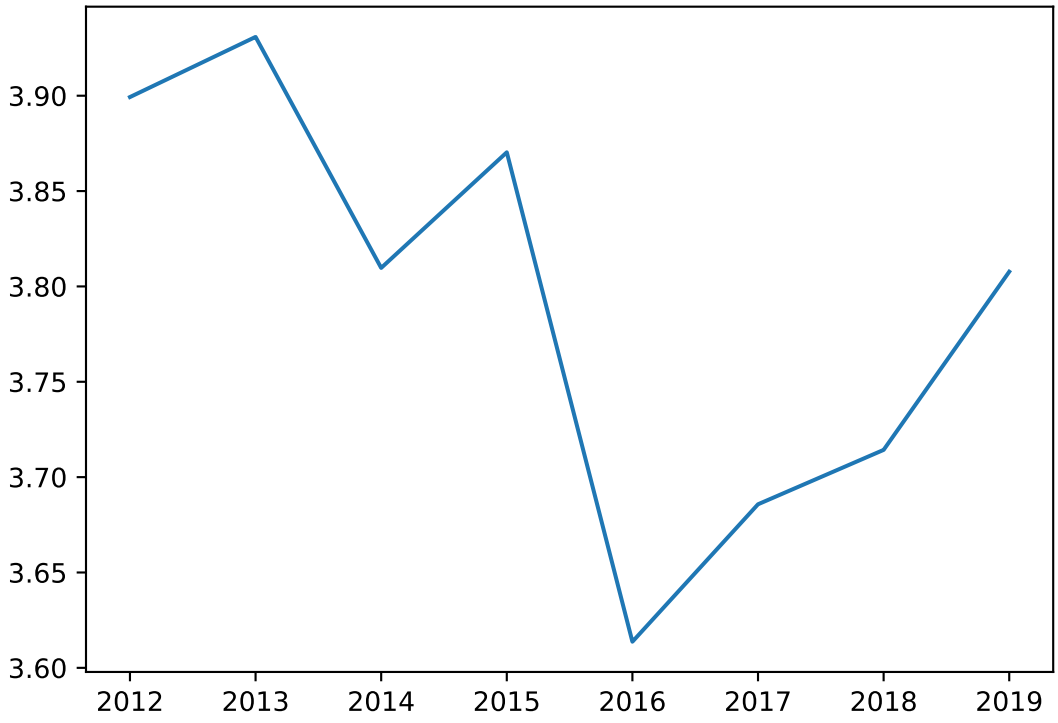

```
visualize_stars_over_time("RESDUcs7fIiihp38-d6_6g", "4JNXUYY8wbaaDmk3BPzlWw", "K7lWdNUhCbcnEvI0NhGewg",
```

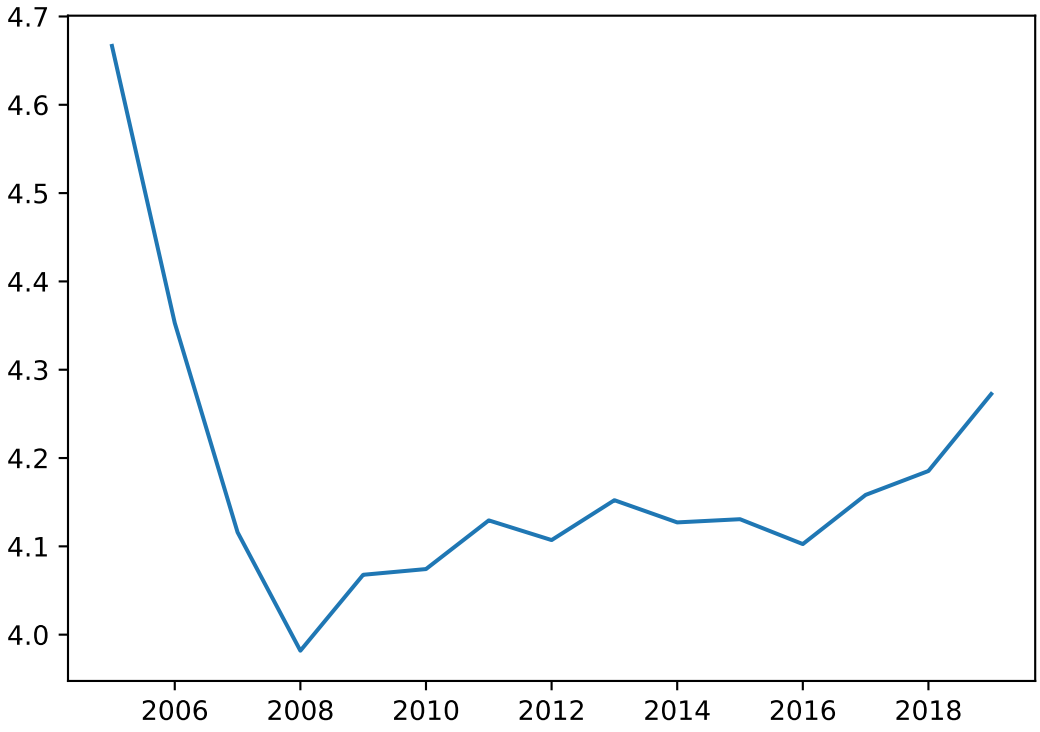

3.9
3.8
3.7
3.6
3.5
2
4
6
8
10
12

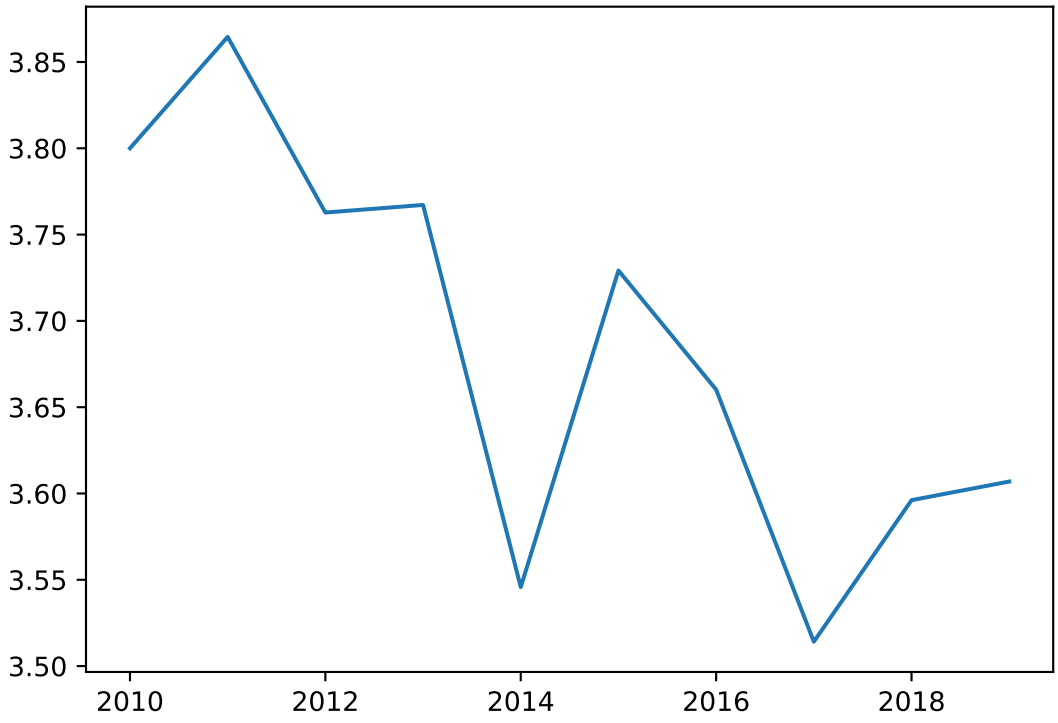

4.200
4.175
4.150
4.125
4.100
4.075
4.050
2
4
6
8
10
12




```
visualize_stars_over_time("RESDUcs7fIiihp38-d6_6g", "4JNXUYY8wbaaDmk3BPzlWw", "K7lWdNUhCbcnEvI0NhGewg",
```

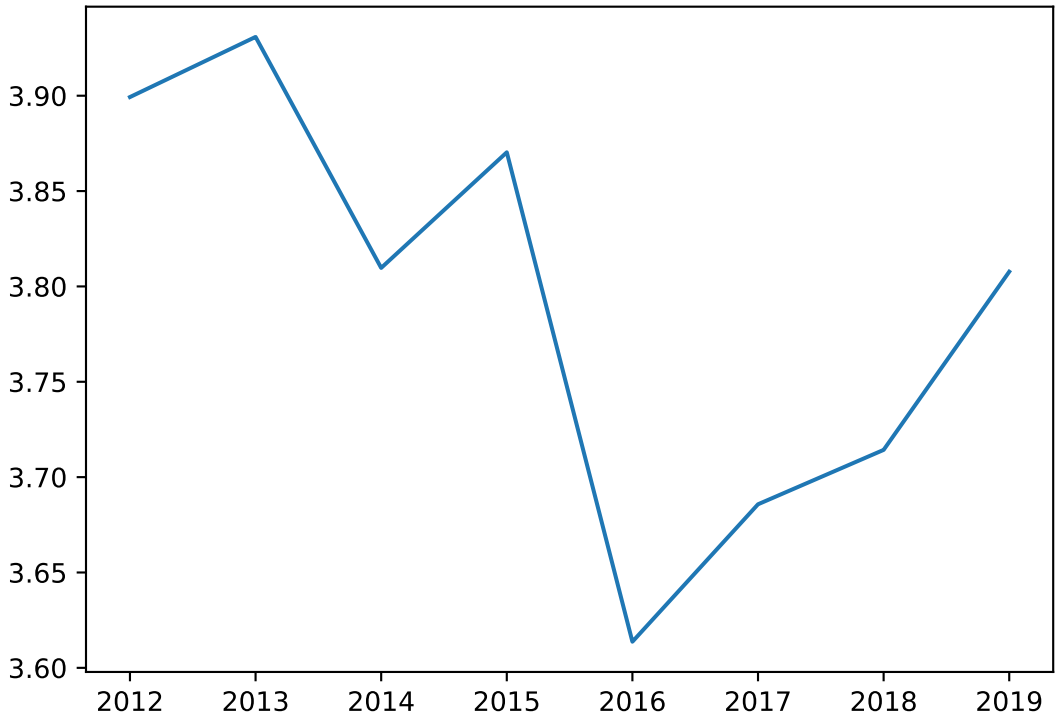

3.90
3.85
3.80
3.75
3.70
3.65
3.60
2012
2013
2014
2015
2016
2017
2018
2019

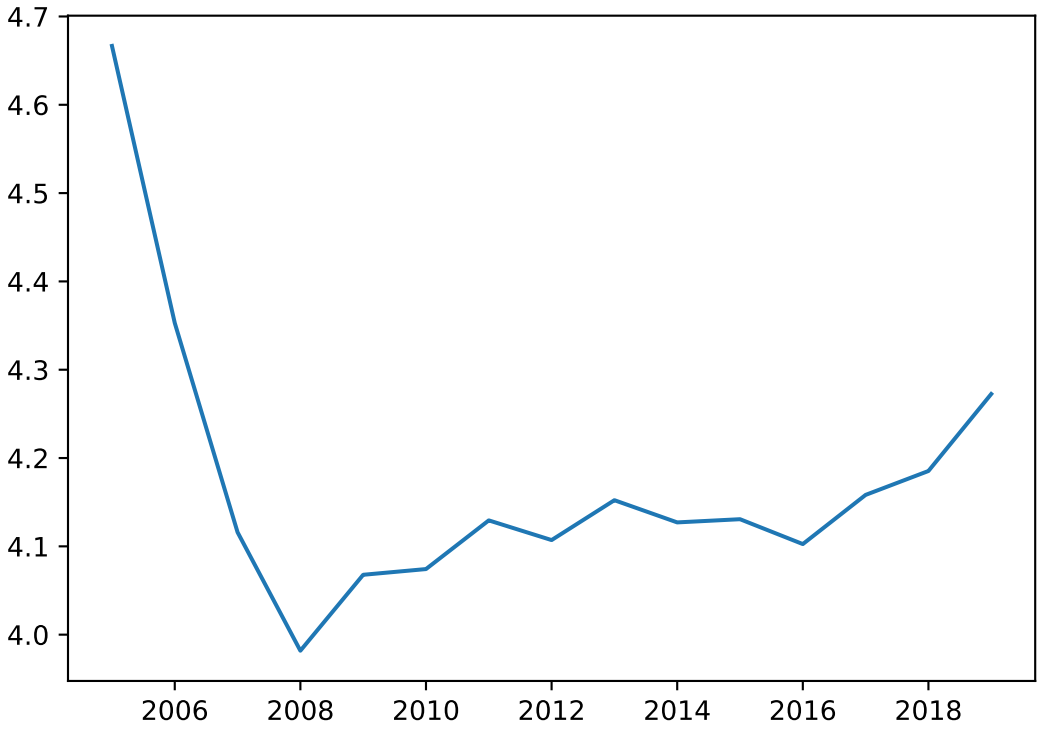

4.7
4.6
4.5
4.4
4.3
4.2
4.1
4.0
2006
2008
2010
2012
2014
2016
2018

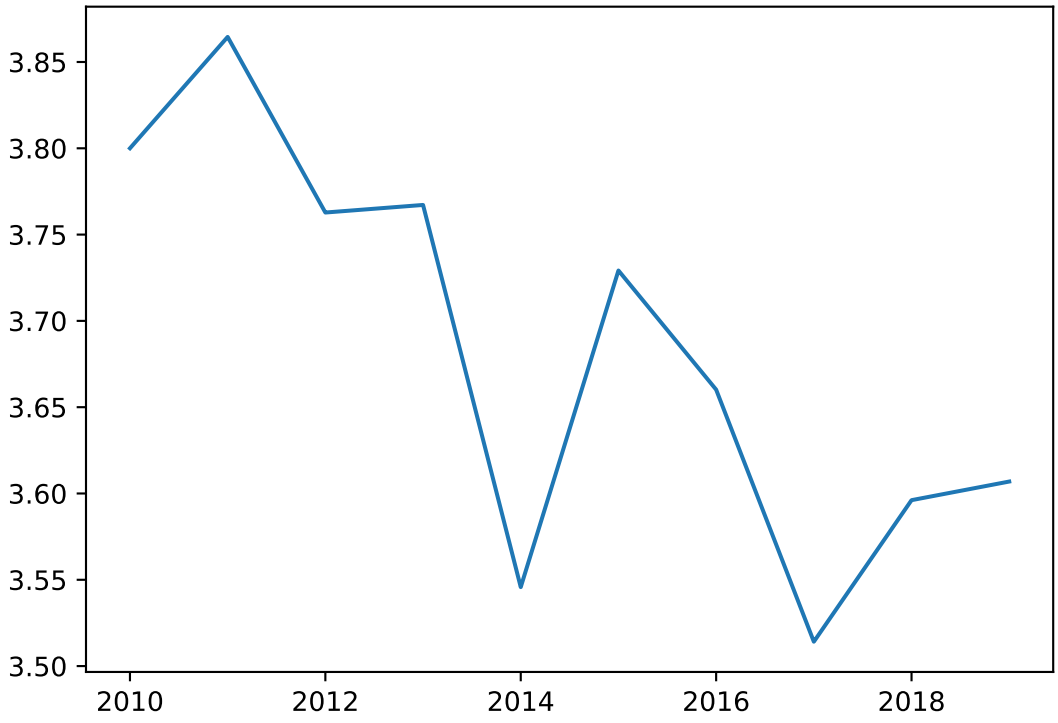


Question 6

```
our_businesses = ["RESDUcs7fIiihp38-d6_6g", "4JNXUYY8wbaaDmk3BPzlWw", "K7lWdNUhCbcnEvI0NhGewg"]

visualize_stars_over_time(*our_businesses, "years")
```


3.90
3.85
3.80
3.75
3.70
3.65
3.60
2012
2013
2014
2015
2016
2017
2018
2019


4.7
4.6
4.5
4.4
4.3
4.2
4.1
4.0
2006
2008
2010
2012
2014
2016
2018



Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.