

```
1 # Week 12 – Assessed exercises
2
3 # This week we learnt some advanced data manipulation methods, about APIs and
4 # about webscraping. In this last set of assessed exercises you must complete the
5 # Brightspace quiz 'W12 – Assessed exercises' and submit a .py file with the
6 # code you used to answer the questions in your quiz. Each question is work
7 # 0.5 marks and it is either correct (full marks) or incorrect (0 marks).
8
9 # This template file contains code that will help you answer the questions in
10 # the quiz.
11
12 # Q1 and Q2 are based on the advanced data manipulation section. You will need to
13 # use the titanic dataset which is part of the seaborn package and can be loaded
14 # using the following commands
15 import seaborn as sb
16
17 titanic = sb.load_dataset('titanic')
18 # The questions involve using the groupby function and applying functions to that
19 # grouped object. For questions involving the interquartile range, use the function
20 # from lecture_12_code.py
21
22 # Q1
23 age_male_sur_mean = round(titanic.age[titanic.sex == 'male'][titanic.survived == 1
24 ].mean())
25 print('age_male_survived_mean:', age_male_sur_mean)
26 # Q2
27 fare_male_sur_std = round(titanic.fare[titanic.sex == 'male'][titanic.survived ==
28 1].std())
```

```
27 print('fare_male_survived_std:', fare_male_sur_std)
28
29 # Q3 to Q5 relate to the World Bank API. You will be asked to search of indicator
30 # and country codes in Q3 and Q4. In Q5 you will need to extract data from the
31 # the World Bank for a particular indicator, country and year
32 import wbddata as wbd
33 from datetime import datetime
34
35
36 # Q3
37 def IQR(x):
38     return x.quantile(0.75) - x.quantile(0.25)
39
40
41 age_female_sur_quan = round(titanic.age[titanic.sex == 'female'][titanic.survived
    == 1].agg(IQR))
42 print('age_female_survived_quartile:', age_female_sur_quan)
43
44 # Q4
45 country_code = wbd.search_countries('Indonesia')[0]['id']
46 print('country code:', country_code)
47
48 # Q5
49 indicator = {'SH.STA.OWAD.ZS': 'prevalence of overweight'}
50 per = round(wbd.get_dataframe(indicator, wbd.search_countries('Ireland')[0]['id']
    ], datetime(2010, 1, 1)).iloc[0][0])
51 print('the prevalence of overweight adults:', per)
52
```

```
53 # Q6 to Q8 relate to webscraping and uses the Spotify weekly charts. You will need
54 # to import BeautifulSoup and the requests package
55 from bs4 import BeautifulSoup
56 import requests
57 import re
58 import numpy as np
59
60 # The below code loads the data from the Spotify weekly charts for the week
61 # 2017-06-30 to 2017-07-07, and uses BeautifulSoup to parse the html.
62 spotify = requests.get('https://spotifycharts.com/regional/global/weekly/2017-06-30--2017-07-07')
63 soup = BeautifulSoup(spotify.text, "html.parser")
64
65 # The following commands extract the information related to the tracks and removes
66 # the html tags
67 track = soup.find_all('td', class_="chart-table-track")
68 tracks = [x.text.strip() for x in track]
69 # Q6 asks you to search through tracks to find the number of times a particular
70 # artist appears in this weekly chart
71 counts = len(
72     [e for e in [re.findall(r'Justin Bieber', tracks[i].replace('\n', ' ')) for i
73                    in range(len(tracks))] if e != []])
74
75 # The following commands extract the information related to the number of plays,
76 # removes the html tags and commas, and converts the value to an integers
77 play = soup.find_all('td', class_="chart-table-streams")
78 plays = [int(x.text.strip().replace(',', '')) for x in play]
```

```
79 # Q7 asks you to perform some statistical analysis on these numbers
80
81 num_plays = round(np.mean(plays))
82 print('means:', num_plays)
83
84 # Q8 asks you to load the charts for a different week and determine how many of
85 # the songs from the original week 2017-06-30 to 2017-07-07 are still in the
86 # charts at this later week. To load in the data for the new week, change the
87 # date range in the url to the date range specified in your question.
88 spotify2 = requests.get('https://spotifycharts.com/regional/global/weekly/2018-06-29--2018-07-06')
89 soup2 = BeautifulSoup(spotify2.text, "html.parser")
90
91 song1 = soup.find_all('strong')
92 songs1 = [x.text for x in song1]
93 song2 = soup2.find_all('strong')
94 songs2 = [x.text for x in song2]
95
96 count = 0
97 for e1 in songs1:
98     for e2 in songs2:
99         if e1 == e2:
100             count += 1
101
102 print('counts:', count)
103
```