Exam information and practice exam

STATS 220 Semester One 2023

Exam information How to revise for the exam!

Information about questions Practice exam Past exams

This practice exam is designed to give you an idea of the format of the exam and the kinds of questions you can expect to be given. **The project details referred to in the practice exam are from Semester One 2022.**

Please use Ed Discussion to post questions related to this practice exam, or general revision related questions 😺

Section 1 Section 2 Section 3

This section will contain questions based on knowledge of web scraping, data manipulations, and automated code-driven processes.

Total 16 marks

Throughout the course, Anna has used data from different weekly charts of the NZ Top 40 singles: **https://nztop40.co.nz/chart/singles (https://nztop40.co.nz/chart/singles)**

Q14

The screenshot below shows the robots.txt file for the NZ Top 40 website.

```
User-agent: *
Disallow: /login
Disallow: /admin
```

Referring specifically to the URL used to scrape songs about singles, discuss whether the robots.txt file indicates that data about songs on the charts should not be web scraped.

2 marks

The robots.txt file says that for all users (this is indicated using *) that the only places that should not be scraped are the URLs that start with "**https://nztop40.co.nz/login (https://nztop40.co.nz/login)**" or "**https://nztop40.co.nz/admin (https://nztop40.co.nz/admin)**". Since the song URLs start with ""**https://nztop40.co.nz/chart (https://nztop40.co.nz/chart)**", the file indicates it's OK to scrape data about songs on the charts.

Q15

The screenshot below shows some of the page source for the web page **https://nztop40.co.nz/chart/singles?chart=5334 (https://nztop40.co.nz/chart/singles?chart=5334)**.

```
<article data-record-id="544741" data-position="1" data-meta-info="First Class by Jac
k Harlow" data-video-id="HmP_wGYw1_g" data-nz="0" class="record_case" style="z-index:
50;">
 <div class="record_wrapper" style="top: 0px;">
  <div class="record_content">
   <div class="record_ribbons">…</div>
   <div class="record_interact">
    <div class="record_label clearfix" title="First Class - Jack Harlow">
     ::before
     <div class="record_number ">
      <p>
       <span>1</span>
      </p>
     </div>
     <h2 class="title">
      <span>FIRST CLASS</span>
     </h2>
     <h3 class="artist">…</h3>
     ::after
    </div>
    <div class="record_buttons">…</div>
    <p class="band_label label_button">…</p>
   </div>
  </div>
  <div class="record_cover">
   <img src="../assets/record_covers/cover_11_1649888698.jpg" alt="First Class by J
   ack Harlow" width="180" height="180">
  </div>
  <img src="../assets/images/touch-icon-white.svg" class="touch-img" alt="icon">
 </div>
</article>
```

Suppose your goal is to scrape the names of each song, the positions each song has on the chart that week, and the URLs for the album covers, to get the resulting `song_names`, `chart_positions` and `album_urls` shown below:

```
> song_names
 [1] "FIRST CLASS"              "N95"
 [3] "AS IT WAS"                "ABOUT DAMN TIME"
 [5] "GO (GODDARD. REMIX)"      "DIE HARD"
 [7] "UNITED IN GRIEF"          "FATHER TIME"
 [9] "COLD HEART (PNAU REMIX)" "HEAT WAVES"
[11] "WAIT FOR U"               "FRIDAY NIGHT"
[13] "STARLIGHT"                "SHIVERS"
[15] "BIG ENERGY"               "BAD HABITS"
[17] "IN THE AIR"               "DOWN UNDER"
[19] "MIDDLE OF THE NIGHT"      "WHERE ARE YOU NOW"
[21] "STAY"                     "FREAKY DEAKY"
[23] "DREAMS"                   "NO ROLE MODELZ"
[25] "ENEMY"                    "COOPED UP"
[27] "DUA LIPA"                 "IN THE STARS"
[29] "THOUSAND MILES"           "INDUSTRY BABY"
[31] "WAIT A MINUTE!"           "BAM BAM"
[33] "LEVITATING"               "GHOST"
[35] "MR REGGAE"                "EASY ON ME"
[37] "COOL IT DOWN"             "WOMAN"
[39] "SAVE YOUR TEARS (REMIX)" "KEEP ON ROLLIN"
```

```
> chart_positions
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
[16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
[31] "31" "32" "33" "34" "35" "36" "37" "38" "39" "40"
```

```
> album_urls
 [1] "../assets/record_covers/cover_11_1649888698.jpg"
 [2] "../assets/record_covers/cover_4_1653003323.jpg"
 [3] "../assets/record_covers/cover_1_1649296712.jpg"
 [4] "../assets/record_covers/cover_0_1650504543.jpg"
 [5] "../assets/record_covers/cover_6_1648688496.jpg"
 [6] "../assets/record_covers/cover_4_1653003323.jpg"
 [7] "../assets/record_covers/cover_4_1653003323.jpg"
 [8] "../assets/record_covers/cover_4_1653003323.jpg"
 [9] "../assets/record_covers/cover_2_1629343631.jpg"
[10] "../assets/record_covers/cover_0_1612507900.jpg"
[11] "../assets/record_covers/cover_6_1651704933.jpg"
[12] "../assets/record_covers/cover_0_1642125276.jpg"
[13] "../assets/record_covers/cover_34_1646863234.jpg"
[14] "../assets/record_covers/cover_0_1631751037.jpg"
[15] "../assets/record_covers/cover_4_1632951843.jpg"
[16] "../assets/record_covers/cover_3_1625094334.jpg"
[17] "../assets/record_covers/cover_0_1625869033.jpg"
[18] "../assets/record_covers/cover_3_1637810792.jpg"
[19] "../assets/record_covers/cover_0_1653022090.jpg"
[20] "../assets/record_covers/cover_23_1628119379.jpg"
[21] "../assets/record_covers/cover_24_1626311474.jpg"
[22] "../assets/record_covers/cover_8_1646257008.jpg"
[23] "../assets/record_covers/cover_0_1602539083.jpg"
[24] "../assets/record_covers/cover_12_1648160774.jpg"
[25] "../assets/record_covers/cover_0_1640312896.jpg"
[26] "../assets/record_covers/cover_5_1653007613.jpg"
[27] "../assets/record_covers/cover_8_1652307494.jpg"
[28] "../assets/record_covers/cover_13_1651705881.jpg"
[29] "../assets/record_covers/cover_23_1651190040.jpg"
[30] "../assets/record_covers/cover_15_1627515327.jpg"
[31] "../assets/record_covers/cover_0_1651208313.jpg"
[32] "../assets/record_covers/cover_5_1646863234.jpg"
[33] "../assets/record_covers/cover_0_1645163459.jpg"
[34] "../assets/record_covers/cover_10_1633563769.jpg"
[35] "../assets/record_covers/cover_6_1638403946.jpg"
[36] "../assets/record_covers/cover_3_1634767818.jpg"
[37] "../assets/record_covers/cover_0_1639094235.jpg"
[38] "../assets/record_covers/cover_2_1625093306.jpg"
[39] "../assets/record_covers/cover_33_1619739890.jpg"
[40] "../assets/record_covers/cover_2_1645131498.jpg"
```

The code below provides code that will accomplish this goal using the package {rvest}, but some parts of the code have been replaced with numbers.

```
page <- {1}("https://nztop40.co.nz/chart/singles?chart=5334")

song_names <- page %>%
  {2}(".title") %>%
  html_text2() %>%
  head({3})

chart_positions <- page %>%
  html_elements("{4}") %>%
  html_text2()

album_urls <- page %>%
  html_elements(".record_cover") %>%
  html_elements("{5}") %>%
  html_attr("{6}")
```

Use the boxes below to enter the missing function, operator, argument name or value.

6 marks

{1} read_html

{2} html_elements

{3} 40

{4} .record_number

{5} img

{6} src

Q16

The value of `album_urls[40]` is
`"../assets/record_covers/cover_2_1645131498.jpg"`

Describe how you could use functions from {stringr} to manipulate the vector `album_urls` to create a new character vector that contains the full URL for each album cover image
e.g. `"https://nztop40.co.nz/assets/record_covers/cover_2_1645131498.jpg"`

Include in your description the specific names of functions you could use and ensure that code approach could be used for any of the weekly top 40 singles charts on the website.

2 marks

I would use `str_replace_all()` to get replace of the ".." from the URL with `"https://nztop40.co.nz"`. Since "." is a special token for regex, I'll need to use `"\\.\\."`. This approach will work for all charts, as the URLs for the album covers are always recorded this way.

The code below is just for you reference and was demonstrated in the lectures - don't submit only code for your answer, you need to write in your own words why and how the different functions will help you

```
album_urls %>%
  str_replace_all("\\.\\.", "https://nztop40.co.nz")
```

Q17

A student wanted to identify the most common words used in the names of songs that feature in the NZ Top 40 charts.

They started by writing the following lines of R code.

```
words <- tibble(song_names) %>%
  separate_rows(song_names, sep = " ") %>%
  count(song_names)
```

Explain the purpose of each line of the following R code as well as the overall goal for the code.

2 marks

The overall goal for the code was to 'tokenise' the song names into words and then find out how many times each word was used in the songs on the charts for that week. The first line of code creates a tibble data frame using the `song_names` vector. The second line manipulates the data frame so that each row contains one of the words

using `separate_rows()` and separating by spaces. The third line using the `count()` function to group by each word and count how many times it was used. The output produced is the data frame called `words` which contains each word and its count.

Q18

Give an example of how the DRY principle could be followed to create a data frame that contains information about the songs on the NZ Top 40 charts during 2021. Focus on the process of iterating over each of the 52 weeks and describe how the functions `map_df()` and `tibble()` could be used.

2 marks

See Lecture 5B.1 and lab task 5B. Essentially, we set up a vector with the reference URLs for each chart, then use `map_df()` to iterate through each URL/week. For each iteration, we grab what we need from the web page for that week's chart, and create a tibble with the data. Anna recommends also recording the chart date for that week, so when all the week's charts are combined you know which chart the songs/rows came from!