Question 1

```
library(httr)
library(tidyverse)
library(jsonlite)
```

Quote Graden (https://github.com/pprathameshmore/QuoteGarden) is an API for about quotes. The website says that it has more than 5000 quotes.

By using that API, do the followings

(a) Get a random quote.

```
#use the endpoint for random quotes
fromJSON("https://quote-garden.herokuapp.com/api/v2/quotes/random")$quote$quoteText
```

```
## [1] "I have done many movies that people hadn't seen. 'The Fountain,' I spent a year
on that. 'The Prestige' with Chris Nolan, and 'Australia.' From my perspective it's very
satisfying. Some movies people see and other movies they don't. 'Wolverine,' 'X Men,' I
know that in some level people know me just for that and it's fine for me."
```

(b) Get all quotes by Albert Einstein. There are how many of them?

(Do not double count identital quotes).

PS 1: there are actually quotes differ only by punctuations, treat them as different quotes for simplicity. PS 2: you will need to use `URLencode` to quote any names with spaces.

```
fromJSON(URLencode(str_glue("https://quote-garden.herokuapp.com/api/v2/authors/Albert Ei
nstein?page=1&limit=NULL")))$quotes %>%
  distinct(quoteText) %>%
  nrow()
```

```
## [1] 139
```

There are 139 unique quotes from Albert Einstein.

(c) Get a random quote in a genre “education”.

Hint: The API does not directly support it, work around it by using loop.

```
n = 0
while( n < 25){
  education_genre = fromJSON("https://quote-garden.herokuapp.com/api/v2/quotes/random")
  n= n+1 # a counter to keep track of how many iterations were used
  if (is.null(education_genre$quote$quoteGenre)) {
    education_genre = fromJSON("https://quote-garden.herokuapp.com/api/v2/quotes/random"
)
  }
  if (education_genre$quote$quoteGenre == "education") {
    break
  }
  n = n-1
}
education_genre
```

```
## $statusCode
## [1] 200
##
## $quote
## $quote$`_id`
## [1] "5eb17aaeb69dc744b4e74d66"
##
## $quote$quoteText
## [1] "Few things are as essential as education."
##
## $quote$quoteAuthor
## [1] "Walter Annenberg"
##
## $quote$quoteGenre
## [1] "education"
##
## $quote$`__v`
## [1] 0
```

(d) Who said the following quote?

```
fromJSON("https://quote-garden.herokuapp.com/api/v2/quotes/performance")
```

```
## $statusCode
## [1] 200
##
## $totalPages
## [1] 7810
##
## $currentPage
## [1] 1
##
## $quotes
##                          _id
## 1  5d91b45d9980192a317c8821
## 2  5d91b45d9980192a317c881e
## 3  5d91b45d9980192a317c8ecd
## 4  5d91b45d9980192a317c97ee
## 5  5d91b45d9980192a317c9a57
## 6  5eb17aadb69dc744b4e70f84
## 7  5eb17aadb69dc744b4e70f36
## 8  5eb17aadb69dc744b4e7101b
## 9  5eb17aadb69dc744b4e7157c
## 10 5eb17aadb69dc744b4e7128b
##
quoteText
## 1
When performance exceeds ambition, the overlap is called success.
## 2
When performance exceeds ambition, the overlap is called success.
## 3
When performance exceeds ambition, the overlap is called success.
## 4
When performance exceeds ambition, the overlap is called success.
## 5
When performance exceeds ambition, the overlap is called success.
## 6
America doesn't reward people of my age, either in day-to-day life or for their performa
nces.
## 7                        The value of old age depends upon the person who reaches it.
To some men of early performance it is useless. To others, who are late to develop, it j
ust enables them to finish the job.
## 8  By age seven, I used to comb my hair for performances, just pull my hair up into a
bun. Granted, it wasn't a very intricate hairstyle. Still, to be that responsible and di
sciplined at age seven is unusual.
## 9                                                    My best kiss was on stage. Kelly R
owland from Destiny's Child gave me a really nice soft kiss on my lips during a performa
nce on my birthday. It was amazing.
## 10                                  Offspring, the due performance on religious rites,
faithful service, highest conjugal happiness and heavenly bliss for the ancestors and on
eself, depend on one's wife alone.
##         quoteAuthor quoteGenre __v
## 1  Cullen Hightower       <NA>  NA
## 2  Cullen Hightower       <NA>  NA
## 3  Cullen Hightower       <NA>  NA
## 4  Cullen Hightower       <NA>  NA
```

```
## 5  Cullen Hightower       <NA>  NA
## 6      Meryl Streep        age   0
## 7      Thomas Hardy        age   0
## 8     Janet Jackson        age   0
## 9       Chris Brown    amazing   0
## 10       Guru Nanak      alone   0
```

> When performance exceeds ambition, the overlap is called success.

Cullen Hightower said “When performance exceeds ambition, the overlap is called success.”.

Hint: the endpoint for searching quotes in missing from the website, but it is on the github repo.

Question 2

In this question, you will be asked to use Yelp API to perform some tasks.

First, you will need to register an app on Yelp platform: https://www.yelp.com/developers/v3/manage_app (https://www.yelp.com/developers/v3/manage_app)

Copy the API key in the file `.Renviron` and do not push it to github!

```
#usethis::edit_r_environ("project")
library(httr)
library(tidyverse)
library(jsonlite)
library(rvest)
library(stringr)
```

(a) Use the "search" endpoint to search for "Burgers and Brew" and get its `id`.

```
r <- GET(
  "https://api.yelp.com/v3/businesses/search",
  add_headers(Authorization = paste("Bearer", Sys.getenv("YELP_TOKEN"))),
  query = list(
    location = "Davis"
  )
)
stop_for_status(r)
json <- content(r, as = "text")
fromJSON(json)$businesses %>% filter(name == "Burgers and Brew") %>% pull(id)
```

```
## [1] "L4e-5b7nyJsdZdi4PREnsQ"
```

The `id` of Davis's Burgers and Brew is "L4e-5b7nyJsdZdi4PREnsQ".

(b) Use the "detail" endpoint to fetch "Burgers and Brew"'s business hour.

```
r <- GET(
  "https://api.yelp.com/v3/businesses/L4e-5b7nyJsdZdi4PREnsQ",
  add_headers(Authorization = paste("Bearer", Sys.getenv("YELP_TOKEN"))),
  query = list(
    location = "Davis"
  )
)
stop_for_status(r)
json <- content(r, as = "text")
fromJSON(json)$hours$open
```

```
## [[1]]
##   is_overnight start  end day
## 1        FALSE  1100 2200   0
## 2        FALSE  1100 2200   1
## 3        FALSE  1100 2200   2
## 4        FALSE  1100 2200   3
## 5        FALSE  1100 2200   4
## 6        FALSE  1100 2200   5
## 7        FALSE  1100 2200   6
```

Burgers and Brew in Davis is open from 11am-10pm everyday of the week.

(c) By using the reviews endpoint, get some reviews for “Burgers and Brew”.

PS: it is a limitation for the free yelp account that only 3 reviews are returned.

```
r <- GET(
  "https://api.yelp.com/v3/businesses/L4e-5b7nyJsdZdi4PREnsQ/reviews",
  add_headers(Authorization = paste("Bearer", Sys.getenv("YELP_TOKEN"))),
  query = list(
    location = "Davis"
  )
)
stop_for_status(r)
json <- content(r, as = "text")
fromJSON(json)$reviews %>% select(text, rating) %>% as.tibble()
```

```
## # A tibble: 3 x 2
##   text                                                                rating
##   <chr>                                                                <int>
## 1 "We were coming back from Tahoe and just so happens this was ooo our s…      4
## 2 "Walked in today for takeout. Cashier/server was nice and courteous, d…      3
## 3 "Been coming here since my college years and it's definitely one of my…      4
```

(d) It is possible to use webscrapping to get more reviews from yelp website directly https://www.yelp.com/biz/burgers-and-brew-davis (https://www.yelp.com/biz/burgers-and-brew-davis). Try to get 40 reviews (user, date, rating and review content) from it.

```
#page 1
html_1 <- read_html("https://www.yelp.com/biz/burgers-and-brew-davis")
#retrieve ratings and user names as they are both under the "aria-label" attribute
ratings_user <- html_1 %>%
  html_nodes("ul") %>%
  html_nodes("li") %>%
  html_nodes("div") %>%
  html_attr("aria-label")
#remove the first element and NA values since they are not useful information
ratings_user <- ratings_user[!is.na(ratings_user)][-1] %>%
  as.tibble()

#logic to seperate our one column containing users and ratings into two distinct columns
#also removes instances of multiples reviews by one user
fixed_df_1 <- ratings_user %>%
  filter(!(str_detect(lag(value), "\\d") & str_detect(value, "\\d"))) %>%
  mutate(ind = rep(c(1, 2),length.out = n())) %>%
  group_by(ind) %>%
  mutate(id = row_number()) %>%
  spread(ind, value) %>%
  select(-id) %>%
  rename(user = 1, rating = 2)

#seperate scrape to retrieve review text and date information using span attribute
review_content_date <- html_1 %>%
  html_node(xpath = "/html/body/div[2]/div[4]/div/div[3]/div/div/div[2]/div[1]/div[3]/se
ction[2]/div[2]/div/ul") %>%
  html_nodes("li") %>%
  html_nodes("span") %>%
  html_text

#vector converted to tibble to make binding all our columns easier later on
#removal of dates associated with multiple entries
dates_1 <- review_content_date[str_detect(review_content_date,"\\d{1,2}/\\d{1,2}/\\d{4}"
)][-c(3, 14, 15, 20)] %>%
  as.tibble() %>%
  rename(dates = value)
#vector converted to tibble to make binding all our columns easier later on
#removal of reviews associated with multiple entries
reviews_1 <- review_content_date[str_detect(review_content_date,".{20}")][-c(3, 11, 15,
16, 21)] %>%
  as.tibble() %>%
  rename(reviews = value)

#page 2
#repetition of the above actions using the additonal parameter start=20 to access the
#second page of reviews
html_2 <- read_html("https://www.yelp.com/biz/burgers-and-brew-davis?start=20")
ratings_user <- html_2 %>%
  html_nodes("ul") %>%
  html_nodes("li") %>%
  html_nodes("div") %>%
  html_attr("aria-label")
```

```
ratings_user <- ratings_user[!is.na(ratings_user)][-1] %>%
  as.tibble()

fixed_df_2 <- ratings_user %>%
  filter(!(str_detect(lag(value), "\\d") & str_detect(value, "\\d"))) %>%
  mutate(ind = rep(c(1, 2),length.out = n())) %>%
  group_by(ind) %>%
  mutate(id = row_number()) %>%
  spread(ind, value) %>%
  select(-id) %>%
  rename(user = 1, rating = 2)

review_content_date <- html_2 %>%
  html_node(xpath = "/html/body/div[2]/div[4]/div/div[3]/div/div/div[2]/div[1]/div[3]/se
ction[2]/div[2]/div/ul") %>%
  html_nodes("li") %>%
  html_nodes("span") %>%
  html_text

dates_2 <- review_content_date[str_detect(review_content_date,"\\d{1,2}/\\d{1,2}/\\d{4}"
)][-c(10, 14, 15)] %>%
  as.tibble() %>%
  rename(dates = value)

reviews_2 <- review_content_date[str_detect(review_content_date,".{20}")][-c(1, 3, 12, 1
6, 17, 20)]%>%
  as.tibble() %>%
  rename(reviews = value)

#bind the rows of our page1&2 tibbles
#then bind all the columns together
melted_df <- bind_cols(bind_rows(fixed_df_1, fixed_df_2), bind_rows(dates_1, dates_2), b
ind_rows(reviews_1, reviews_2))

#our end result is a 40 x 4 tibble containing information on
#users, ratings, dates, reviews (addtional reviews from the same user have been omitted)
melted_df
```

```
## # A tibble: 40 x 4
##    user       rating     dates   reviews
##    <chr>      <chr>      <chr>   <chr>
##  1 Mishan G.  4 star ra… 2/1/20… We were coming back from Tahoe and just so hap…
##  2 Sanghoo P. 3 star ra… 5/17/2… Walked in today for takeout. Cashier/server wa…
##  3 Kelly H.   4 star ra… 2/1/20… Been coming here since my college years and it…
##  4 Julia S.   4 star ra… 2/3/20… Smoked aged cheddar burger (lettuce for bun) w…
##  5 Abner P.   1 star ra… 5/17/2… Had not been at burgers and brew for a while a…
##  6 John A.    5 star ra… 5/28/2… A good place to go with family or with friends…
##  7 Katherine… 4 star ra… 2/28/2… I've been to Blast and Brew twice and each tim…
##  8 Suzie L.   5 star ra… 12/2/2… Burgers and Brew is the best burger place in D…
##  9 Maribel M. 5 star ra… 3/3/20… The burgers are huge and good! I enjoyed my cu…
## 10 Yuxin W.   1 star ra… 5/14/2… We order from them regularly on Doordash and w…
## # … with 30 more rows
```

PS: you only need static web scrapping. Remark: Do q3 first before attemping this question. It is not easy because yelp is avoiding user to “inpect” the source code.

Question 3

```
library(tidyverse)
library(rvest)
library(stringr)
```

(a) By visiting https://statistics.ucdavis.edu/courses/descriptions-undergrad (https://statistics.ucdavis.edu/courses/descriptions-undergrad), scrape the course information including course numbers, titles, units and descriptions.

Make a dataframe out of it.

The end result should look identical to:

```
#> # A tibble: 36 x 4
#>    course  title                     unit  description
#>    <chr>   <chr>                     <chr> <chr>
#>  1 STA 010 Statistical Thinking      4     Lecture—3 hour(s); Discussion/Labora…
#>  2 STA 012 Introduction to Discrete… 4     Lecture—3 hour(s); Laboratory—1 hour…
#>  3 STA 013 Elementary Statistics     4     Lecture—3 hour(s); Discussion—1 hour…
#>  4 STA 01… Elementary Statistics     4     Lecture—1.5 hour(s); Web Virtual Lec…
#>  5 STA 032 Gateway to Statistical D… 4     Lecture—3 hour(s); Laboratory—1 hour…
#>  6 STA 09… Seminar                   1-2   Seminar—1-2 hour(s). Prerequisite(s)…
#>  7 STA 098 Directed Group Study      1-5   Variable. Prerequisite(s): Consent o…
#>  8 STA 099 Special Study for Underg… 1-5   Variable. Prerequisite(s): Consent o…
#>  9 STA 100 Applied Statistics for B… 4     Lecture—3 hour(s); Laboratory—1 hour…
#> 10 STA 101 Advanced Applied Statist… 4     Lecture—3 hour(s); Laboratory—1 hour…
#> # … with 26 more rows
```

```
#html link to read
html <- read_html("https://statistics.ucdavis.edu/courses/descriptions-undergrad")
#find courses using the h2 tag
courses <- html %>%
  html_nodes("h2") %>%
  html_text()
#find descriptions using the p tag
descriptions <- html %>%
  html_nodes("p") %>%
  html_text()
#create tibble from information scrapped.
#use regex to seperate course into 3 distinct columns: course, title, unit
answer_df <- tibble(course = courses[-c(1,2, 39, 40)], descriptions = descriptions[-c(37
, 38)]) %>%
  separate(course, c("course", "title", "unit"), sep = "([—,(,)])")
answer_df
```

```
## # A tibble: 36 x 4
##    course  title                     unit  descriptions
##    <chr>   <chr>                     <chr> <chr>
##  1 STA 010 Statistical Thinking      4     Lecture—3 hour(s); Discussion/Labora…
##  2 STA 012 Introduction to Discrete… 4     Lecture—3 hour(s); Laboratory—1 hour…
##  3 STA 013 Elementary Statistics     4     Lecture—3 hour(s); Discussion—1 hour…
##  4 STA 01… Elementary Statistics     4     Lecture—1.5 hour(s); Web Virtual Lec…
##  5 STA 032 Gateway to Statistical D… 4     Lecture—3 hour(s); Laboratory—1 hour…
##  6 STA 09… Seminar                   1-2   Seminar—1-2 hour(s). Prerequisite(s)…
##  7 STA 098 Directed Group Study      1-5   Variable. Prerequisite(s): Consent o…
##  8 STA 099 Special Study for Underg… 1-5   Variable. Prerequisite(s): Consent o…
##  9 STA 100 Applied Statistics for B… 4     Lecture—3 hour(s); Laboratory—1 hour…
## 10 STA 101 Advanced Applied Statist… 4     Lecture—3 hour(s); Laboratory—1 hour…
## # … with 26 more rows
```

(b) By visiting https://statistics.ucdavis.edu/courses/expanded-descriptions (https://statistics.ucdavis.edu/courses/expanded-descriptions), scrape all the links for lower and upper division courses.

```
html <- read_html("https://statistics.ucdavis.edu/courses/expanded-descriptions")
#use href attribute of a tag to find the extensions for each course
course_url <- html %>%
  html_node("article") %>%
  html_nodes("a") %>%
  html_attr("href")

#concatenate the statistics link to the extensions found to form valid urls
courses <- tibble(courses = paste0("https://statistics.ucdavis.edu", course_url[c(1:27
)]))
courses
```

```
## # A tibble: 27 x 1
##    courses
##    <chr>
##  1 https://statistics.ucdavis.edu/expanded-descriptions/10
##  2 https://statistics.ucdavis.edu/expanded-descriptions/12
##  3 https://statistics.ucdavis.edu/expanded-descriptions/13
##  4 https://statistics.ucdavis.edu/expanded-descriptions/10
##  5 https://statistics.ucdavis.edu/expanded-descriptions/32
##  6 https://statistics.ucdavis.edu/expanded-descriptions/100
##  7 https://statistics.ucdavis.edu/expanded-descriptions/101
##  8 https://statistics.ucdavis.edu/expanded-descriptions/103
##  9 https://statistics.ucdavis.edu/expanded-descriptions/104
## 10 https://statistics.ucdavis.edu/expanded-descriptions/106
## # … with 17 more rows
```

(c) By using the links from (b), extracts all the prerequisite of the courses and join with the result in (a).

```
#create empty tibble that we're going to add data to using a for loop
prereqs <- tibble(prereq = character())

#visit each of the links in our `courses` tibble
#extract information from the xpath pointing to the prerequisite paragraph (p tag)
#add a row containing the prequisite information for that course
for (i in courses$courses) {
  html <- read_html(i)
  prereqs <- rbind(prereqs, html %>%
    html_node(xpath = "/html/body/div/div/main/div[2]/section/div/div/div/article/div/di
v/p[4]") %>%
    html_text() %>%
    tibble(prereq = .))
}

#bind together the columns of our tibble in (b) and our `prereq` tibble to
#generate a "key" column for our full join
mixed_df <- bind_cols(tibble(course = answer_df$course[-c(6:8, 31:36)]),prereqs)

#join tibble from (a) and (c)
full_join(answer_df, mixed_df, by = c("course"))
```

```
## # A tibble: 36 x 5
##    course  title             unit  descriptions               prereq
##    <chr>   <chr>             <chr> <chr>                      <chr>
##  1 STA 010 Statistical Thin… 4     Lecture—3 hour(s); Discus… Prerequisite: Two…
##  2 STA 012 Introduction to … 4     Lecture—3 hour(s); Labora… Prerequisite: Two…
##  3 STA 013 Elementary Stati… 4     Lecture—3 hour(s); Discus… Prerequisite: two…
##  4 STA 01… Elementary Stati… 4     Lecture—1.5 hour(s); Web … Prerequisite: Two…
##  5 STA 032 Gateway to Stati… 4     Lecture—3 hour(s); Labora… Prerequisite: MAT…
##  6 STA 09… Seminar           1-2   Seminar—1-2 hour(s). Prer… <NA>
##  7 STA 098 Directed Group S… 1-5   Variable. Prerequisite(s)… <NA>
##  8 STA 099 Special Study fo… 1-5   Variable. Prerequisite(s)… <NA>
##  9 STA 100 Applied Statisti… 4     Lecture—3 hour(s); Labora… Prerequisite: Mat…
## 10 STA 101 Advanced Applied… 4     Lecture—3 hour(s); Labora… Prerequisite: cou…
## # … with 26 more rows
```