Extracting data from APIs

Exercises 1. Go to rapidapi.com and create a user

Nokia acquires Rapid technology and team! 🚀

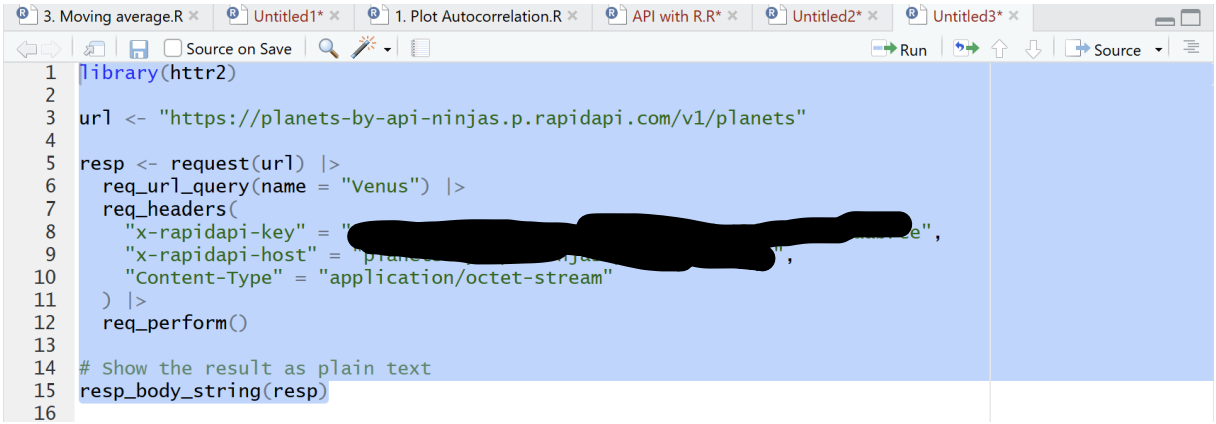
2. Search for and enter into the “Planets by API-Ninjas” API

Under “Pricing” subscribe to the “basic” (free) plan

Go to “Endpoints”

Planets by API-Ninjas

3. Use your own API key for “Planets by API-Ninjas”, and make a request with the parameter “name” set to “Venus”.




```r
library(httr2)

url <- "https://planets-by-api-ninjas.p.rapidapi.com/v1/planets"

resp <- request(url) |>
  req_url_query(name = "Venus") |>
  req_headers(
    "x-rapidapi-key" = "[redacted]",
    "x-rapidapi-host" = "[redacted]",
    "Content-Type" = "application/octet-stream"
  ) |>
  req_perform()

# Show the result as plain text
resp_body_string(resp)
```

4. Modify the R code

```r
req <- request("http://165.22.92.178:8080") %>%
  req_url_path("responses") %>%
  req_url_query(format = "string") %>%
  req_headers(authorization = "DM_DV_123#!")

response <- req %>%
  req_perform()
response # Inspect content type
response %>%
  resp_body_string()
```

Such that you get :

a response in json (by changing the value of the “format” parameter) and interpret the response correctly.

get a response in html and interpret the response correctly.

```r
# Change to json
library(httr2)

req <- request("http://165.22.92.178:8080") |>
  req_url_path("responses") |>
  req_url_query(format = "json") |>
  req_headers(authorization = "DM_DV_123#!")

response <- req %>%
  req_perform ()

response

response %>%
  resp_body_json ()
```

```r
#Change to html
library(httr2)

req <- request("http://165.22.92.178:8080") |>
  req_url_path("responses") |>
  req_url_query(format = "html") |>
  req_headers(authorization = "DM_DV_123#!")

response <- req %>%
  req_perform ()

response

response %>%
  resp_body_string ()
```

5. Send an API request with the following specifications

   URL: "http://165.22.92.178:8080"

   Path: “lm”

   Include a header where “authorization” is set to the value "DM_DV_123#!“

   Include a request body in json format. The body should be specified as follows

      Simulate data as in slide 17, but now only include y, x1, and x2 in the dataframe, and simulate with only 70 observations.

      Give the dataframe to req_body_json(), but remember to transform the dataframe to a named list using as.list().

   Confirm the response (which is delivered in json format) gives three numbers.

```r
# Exercise 5
n <- 70
x1 <- rnorm(n = n)
x2 <- rnorm(n = n)
y <- 2*x1 + 3*x2 + rnorm(n = n)
df <- round(data.frame(y = y, x1 = x1, x2 = x2))
# Construct a request including the data
req <- request("http://165.22.92.178:8080") %>%
  req_url_path("lm") %>%
  req_body_json(as.list(df)) %>%
  req_headers(authorization = "DM_DV_123#!")
# Send the request to the API
resp <- req %>%
  req_perform()
# View the result
resp %>%
  resp_body_json()
```

6. Subscribe to the Google translate API from RapiAPI
Google Translate

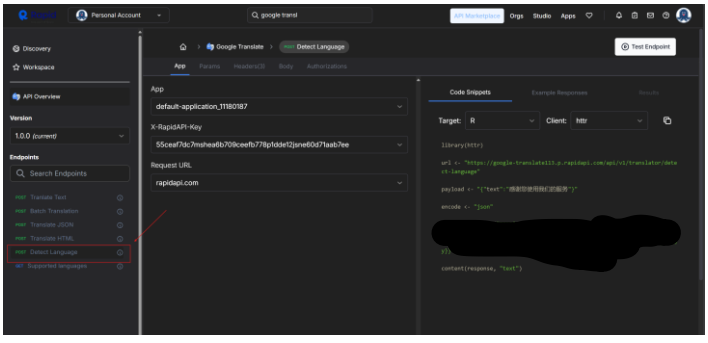
7. In the Google translate API utilize the endpoint “detect”.
   Write a sentence in your native language and let the “detect” endpoint determine the language of your sentence.
   Inspect the response to your request: Which language did Google translate detect, and what is the trust level?

```
> # Set your RapidAPI key
> api_key <- "55ceaf7dc7mshea6b709ceefb778p1dde12jsne60d71aab7ee"
>
> # Define the URL for the text translation endpoint
> url <- "https://google-translate113.p.rapidapi.com/api/v1/translator/text"
>
> # Define the sentence you want to translate
> sentence <- "Hallo, wie geht es dir?"
>
> # Build the payload
> payload <- list(
+   from = "auto",    # auto-detect the language
+   to = "en",        # translate to English
+   text = sentence
+ )
>
> # Build and send the request
> resp <- request(url) |>
+   req_headers(
+     [redacted],
+     "Content-Type" = "application/json"
+   ) |>
+   req_body_json(payload) |>
+   req_perform()
>
> # Inspect the raw JSON response
> result <- resp |> resp_body_json()
>
> # Print the full response structure
> print(result)
$trans
[1] "Hello, how are you?"

$source_language_code
[1] "de"

$source_language
[1] "German"

$trust_level
[1] 1
```

Here because we have set language from to auto it automaticcly gives source_language and trust_level

Or

Got to rapidapi and select detect




Copy the code and ask chatgpt to change it from httr to httr2 and paste it into R