HTTP

INTERMEDIATE IMPORTING DATA IN R

Filip Schouwenaars
Instructor, DataCamp

datacamp

Data on the web

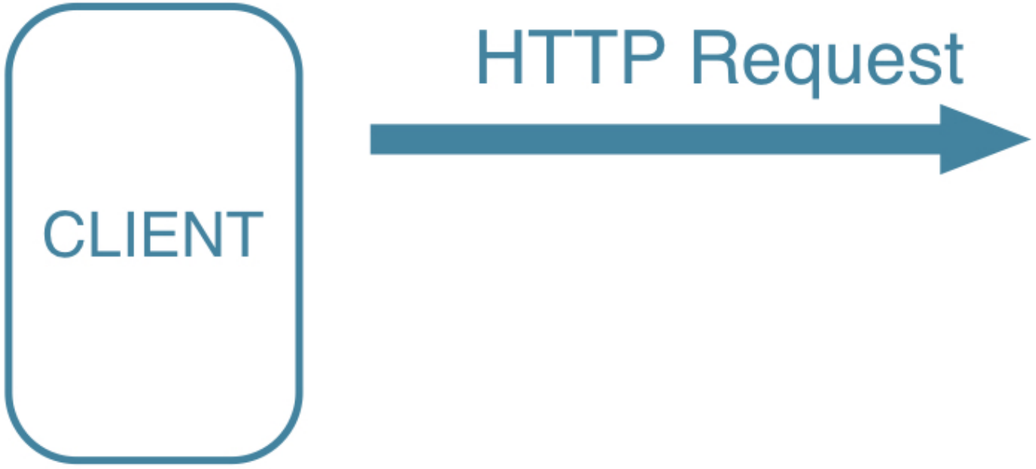
- Already worked with it!
- Many packages handle it for you
- File formats useful for web technology
- JSON

HTTP

- HyperText Transfer Protocol
- Rules about data exchange between computers
- Language of the web

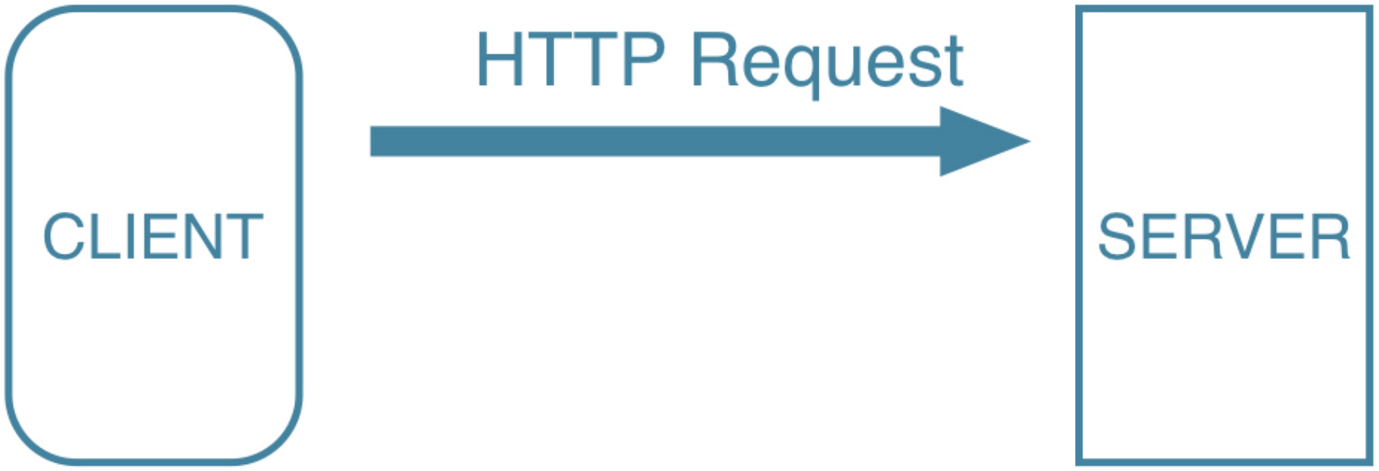
HTTP

- HyperText Transfer Protocol
- Rules about data exchange between computers
- Language of the web



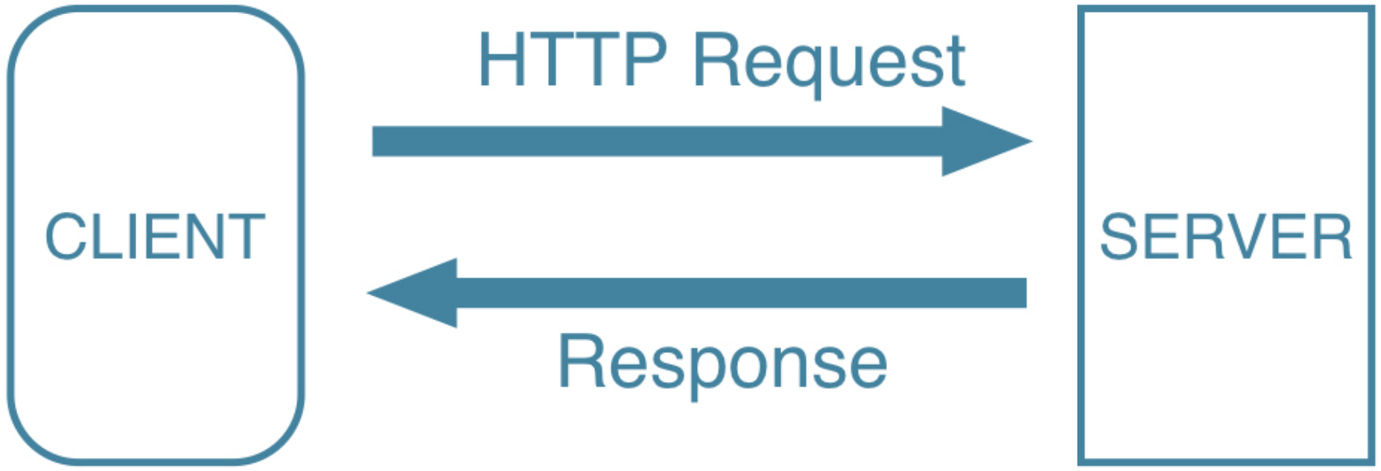
HTTP

- HyperText Transfer Protocol
- Rules about data exchange between computers
- Language of the web



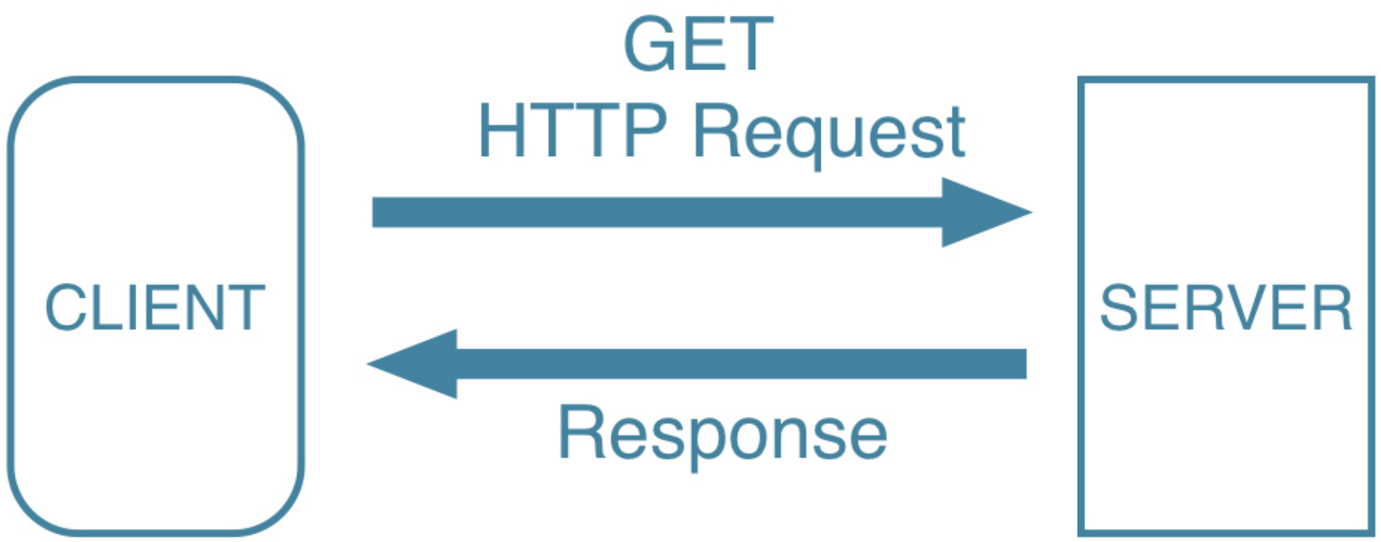
HTTP

- HyperText Transfer Protocol
- Rules about data exchange between computers
- Language of the web



HTTP

- HyperText Transfer Protocol
- Rules about data exchange between computers
- Language of the web



Example: CSV

http://s3.amazonaws.com/ ... /states.csv

```
# Manually download file through web browser

read.csv(url("path/to/states.csv"))
```

```
         state    capital pop_mill area_sqm
1 South Dakota     Pierre    0.853    77116
2     New York     Albany   19.746    54555
3       Oregon      Salem    3.970    98381
4      Vermont Montpelier    0.627     9616
5       Hawaii   Honolulu    1.420    10931
```

Example: CSV

```
read.csv("http://s3.amazonaws.com/
                 assets.datacamp.com/course/
                 importing_data_into_r/states.csv")
```

```
         state    capital pop_mill area_sqm
1 South Dakota     Pierre    0.853    77116
2     New York     Albany   19.746    54555
3       Oregon      Salem    3.970    98381
4      Vermont Montpelier    0.627     9616
5       Hawaii   Honolulu    1.420    10931
```

- R sees it's a URL, does GET request, and reads in the CSV file

Example: CSV

```
read.csv("https://s3.amazonaws.com/
                  assets.datacamp.com/course/
                  importing_data_into_r/states.csv")
```

```
         state    capital pop_mill area_sqm
1 South Dakota     Pierre    0.853    77116
2     New York     Albany   19.746    54555
3       Oregon      Salem    3.970    98381
4      Vermont Montpelier    0.627     9616
5       Hawaii   Honolulu    1.420    10931
```

- HTTPS support since R version 3.2.2

Downloading files

INTERMEDIATE IMPORTING DATA IN R

Filip Schouwenaars
Instructor, DataCamp

datacamp

Example: Excel

```r
library(readxl)
read_excel("http://s3.amazonaws.com/
                 assets.datacamp.com/course/
                 importing_data_into_r/cities.xlsx")
```

```
Error:
'http://s3.amazonaws.com/assets.datacamp.com/course/importing_data_into_r/cities.xlsx'
does not exist in current working directory.
```

download.file()

```
url <- "http://s3.amazonaws.com/assets.datacamp.com/
            course/importing_data_into_r/cities.xlsx"
dest_path <- file.path("~", "local_cities.xlsx")
download.file(url, dest_path)
```

```
// Messages showing download progress omitted
```

```
read_excel(dest_path)
```

```
    Capital Population
1  New York   16044000
2    Berlin    3433695
3    Madrid    3010492
4 Stockholm    1683713
```

Why download.file()?

- Reproducibility
- HTTP from inside R
  - Authentication
  - Additional parameters
  - httr - Hadley Wickham

APIs & JSON

INTERMEDIATE IMPORTING DATA IN R

Filip Schouwenaars
Instructor, DataCamp

datacamp

Other data formats

- Before: pages and files from the web
- JSON
- Simple, concise, well-structured
- Human-readable
- Easy to parse and generate for computers
- For communication with Web APIs

API

- Application Programming Interface
- Set of routines and protocols for building software
- How different components interact
- Web API
  - interface to get or add data to server
  - HTTP verbs (GET and others)

Twitter

- https://dev.twitter.com/rest/public
- Get tweets
- Place comments on tweets
- Many applications
  - Research effect of tweets

Info on Rain Man (1988)

```r
url <- "http://www.imdb.com/title/tt0095953/"
download.file(url, "local_imdb.html")
```

```html
        <div class="pro-title-link text-center">
<a href="http://pro.imdb.com/title/tt0095953?rf=cons_tt_contact&ref_=cons_tt_conta
>Contact the Filmmakers on IMDbPro &raquo;</a>
        </div> </td>
                    <td id="overview-top">
        <div id="prometer_container">
            <div id="prometer" class="meter-collapsed up">
                <div id="meterHeaderBox">
                    <div id="meterTitle" class="meterToggleOnHover">Popularity</di
                    <span id="meterRank">1,303</span>
                </div>
                <div id="meterChangeRow" class="meterToggleOnHover">
                        <span>Up</span>
                    <span id="meterChange">163</span>
                    <span>this week</span>
                </div>
            </div>
        </div>
<h1 class="header"> <span class="itemprop" itemprop="name">Rain Man</span>
            <span class="nobr">(<a href="/year/1988/?ref_=tt_ov_inf"
>1988</a>)</span>
```

Rain Man JSON (OMDb API)

http://www.omdbapi.com/?i=tt0095953&r=json

www.omdbapi.com/?i=tt0095953&r=json


```
{
    Title: "Rain Man",
    Year: "1988",
    Rated: "R",
    Released: "16 Dec 1988",
    Runtime: "133 min",
    Genre: "Drama",
    Director: "Barry Levinson",
    Writer: "Barry Morrow (story), Ronald Bass (screenplay), Barry Morrow (screenplay)",
    Actors: "Dustin Hoffman, Tom Cruise, Valeria Golino, Gerald R. Molen",
    Plot: "Selfish yuppie Charlie Babbitt's father left a fortune to his savant brother
    Raymond and a pittance to Charlie; they travel cross-country.",
    Language: "English, Italian",
    Country: "USA",
    Awards: "Won 4 Oscars. Another 23 wins & 19 nominations.",
    Poster: "http://ia.media-
    imdb.com/images/M/MV5BMTQ4NTA1NDU3NV5BMl5BanBnXkFtZTcwODUwMTU2NA@@._V1_SX300.jpg",
    Metascore: "65",
    imdbRating: "8.0",
    imdbVotes: "362,306",
    imdbID: "tt0095953",
    Type: "movie",
    Response: "True"
}
```

jsonlite

- Jeroen Ooms
- Improvement of earlier packages
- Consistent, robust
- Support all use-cases

Rain Man list in R

```r
install.packages("jsonlite")
library(jsonlite)
fromJSON("http://www.omdbapi.com/?i=tt0095953&r=json")
```

```
List of 20
 $ Title     : chr "Rain Man"
 $ Year      : chr "1988"
 $ Rated     : chr "R"
 $ Released  : chr "16 Dec 1988"
 $ Runtime   : chr "133 min"
 ...
 $ imdbVotes : chr "359,903"
 $ imdbID    : chr "tt0095953"
 $ Type      : chr "movie"
 $ Response  : chr "True"
```

- Way more structure!

JSON & jsonlite

INTERMEDIATE IMPORTING DATA IN R

Filip Schouwenaars
Instructor, DataCamp

datacamp

JSON object

```
{"id":1,"name":"Frank","age":23,"married":false}
```

name	value
string	string
	number
	boolean
	null
	JSON object
	JSON array

JSON object

```
{"id":1,"name":"Frank","age":23,"married":false}
```

```
x <- '{"id":1,"name":"Frank","age":23,"married":false}'
r <- fromJSON(x)
str(r)
```

```
List of 4
 $ id     : int 1
 $ name   : chr "Frank"
 $ age    : int 23
 $ married: logi FALSE
```

JSON array

```
JSON
[4, 7, 4, 6, 4, 5, 10, 6, 6, 8]
```

```
fromJSON('[4, 7, 4, 6, 4, 5, 10, 6, 6, 8]')
```

```
4  7  4  6  4  5 10  6  6  8
```

```
JSON
[4, "a", 4, 6, 4, "b", 10, 6, false, null]
```

```
fromJSON('[4, "a", 4, 6, 4, "b", 10, 6, false, null]')
```

```
"4"  "a"  "4"  "6"  "4"  "b"  "10"  "6"  "FALSE"  NA
```

JSON Nesting

```
{
    "id": 1,
    "name": "Frank",
    "age": 23,
    "married": false,
    "partner": {
      "id": 4,
      "name": "Julie"
    }
}
```

JSON Nesting

```
r <- fromJSON('{"id":1,"name":"Frank","age":23,
      "married":false,"partner":{"id":4,"name":"Julie"}}')
Rstr(r)
```

```
List of 5
 $ id     : int 1
 $ name   : chr "Frank"
 $ age    : int 23
 $ married: logi FALSE
 $ partner:List of 2
  ..$ id  : int 4
  ..$ name: chr "Julie"
```

JSON Array of JSON Objects

```
[
    {"id":1, "name":"Frank"},
    {"id":4, "name":"Julie"},
    {"id":12, "name":"Zach"}
]
```

```
RfromJSON('[{"id":1, "name":"Frank"},
            {"id":4, "name":"Julie"},
            {"id":12, "name":"Zach"}]')
```

```
  id  name
1  1 Frank
2  4 Julie
3 12  Zach
```

Other jsonlite functions

- toJSON()
- prettify()
- minify()