Using `XPATH` for web scraping

Example 3. Product price scraping from https://nepalfoods.gov.np/

In [ ]:
```
library(rvest) #see https://rvest.tidyverse.org/articles/harvesting-the-web.html for details
library(dplyr)

#loading webpage content
webpage <- read_html("https://nepalfoods.gov.np/")
```

In [2]:
```
category <- webpage %>% html_nodes(xpath = "//div[@class='product-category']") %>% html_text(trim=TRUE)
length(category)
head(category)

product <- webpage %>% html_nodes(xpath = "//h2") %>% html_text(trim=TRUE)
length(product)
head(product)

price <- webpage %>% html_nodes(xpath = "//div[@class='product-price']") %>% html_text(trim=TRUE)
length(price)
head(price)
```

```
27
'अन्य' · 'अन्य' · 'चामल' · 'चामल' · 'चामल' · 'दाल'

27
'उवा १ केजी' · 'चियापत्ती ५०० ग्राम' · 'Long Grain चामल  १० केजी' · 'हुम्लाको कागुनोको चामल १ केजी' · 'हुम्लाको चिनोको चामल १ केजी' · 'कर्णालीको सिमि १ केजी'

27
'NPR 200.00' · 'NPR 270.00' · 'NPR 1780.00' · 'NPR 260.00' · 'NPR 260.00' · 'NPR 240.00'
```

In [3]:
```
#creating dataframe from category, product, and price list
df <- data.frame(category, product, price)
df <- df %>% arrange(category)
head(df)

#saving dataframe
write.csv(df, file = "example3.csv", row.names=FALSE)
```

A data.frame: 6 × 3

	category	product	price
	<chr>	<chr>	<chr>
1	अन्य	उवा १ केजी	NPR 200.00
2	अन्य	चियापत्ती ५०० ग्राम	NPR 270.00
3	अन्य	टाइमपास टाइचिन चिउरा १ केजी	NPR 100.00
4	अन्य	गहुँ आटा 5 केजी	NPR 360.00
5	अन्य	डी.डी.सी डेरी घ्यू १/२ लि	NPR 580.00
6	अन्य	STC ग्यास सिलिण्डर(Exchange only STC Cylinder)	NPR 1910.00

Example 4. Extract information on Top Box Office movies from https://www.imdb.com/chart/boxoffice

In [4]:
```
#loading webpage content
webpage <- read_html("https://www.imdb.com/chart/boxoffice")
```

In [5]:
```
movie <- webpage %>% html_nodes(xpath = "//a[@class='ipc-title-link-wrapper']") %>% html_text(trim=TRUE)
length(movie)
print(movie)

weekend_gross <- webpage %>% html_nodes(xpath = "//span[contains(.,'Weekend Gross:')]/parent::*/span[2]") %>% html_text(trim=T
length(weekend_gross)
print(weekend_gross)

total_gross <- webpage %>% html_nodes(xpath = "//span[contains(.,'Total Gross:')]/parent::*/span[2]") %>% html_text(trim=TRUE)
length(total_gross)
print(total_gross)

weeks_released <- webpage %>% html_nodes(xpath = "//span[contains(.,'Weeks Released:')]/parent::*/span[2]") %>% html_text(trim
length(weeks_released)
print(weeks_released)

rating <- webpage %>% html_nodes(xpath = "//span[@data-testid='ratingGroup--imdb-rating']") %>% html_text(trim=TRUE)
length(rating)
print(rating)
```

10

```
 [1] "1. Inside Out 2"
 [2] "2. Bad Boys: Ride or Die"
 [3] "3. Kingdom of the Planet of the Apes"
 [4] "4. The Garfield Movie"
 [5] "5. IF"
 [6] "6. The Watchers"
 [7] "7. Furiosa: A Mad Max Saga"
 [8] "8. The Fall Guy"
 [9] "9. The Strangers: Chapter 1"
[10] "10. The Lord of the Rings: The Fellowship of the Ring"
```

10

```
 [1] "$154M" "$34M"  "$5.5M" "$4.8M" "$3.6M" "$3.5M" "$2.6M" "$1.6M" "$759K"
[10] "$633K"
```

10

```
 [1] "$154M" "$113M" "$158M" "$78M"  "$101M" "$14M"  "$63M"  "$88M"  "$34M"
[10] "$319M"
```

10

```
 [1] "1" "2" "6" "4" "5" "2" "4" "7" "5" "2"
```

10

```
 [1] "8.0 (16K)"  "7.0 (19K)"  "7.2 (53K)"  "5.8 (8.4K)" "6.7 (14K)"
 [6] "5.8 (6.9K)" "7.8 (85K)"  "7.0 (101K)" "4.7 (11K)"  "8.9 (2M)"
```

In [6]:
```
df <- data.frame(movie, weekend_gross, total_gross, weeks_released, rating)
df

write.csv(df, file = 'example4.csv', row.names=FALSE)
```

A data.frame: 10 × 5

movie	weekend_gross	total_gross	weeks_released	rating
<chr>	<chr>	<chr>	<chr>	<chr>
1. Inside Out 2	$154M	$154M	1	8.0 (16K)
2. Bad Boys: Ride or Die	$34M	$113M	2	7.0 (19K)
3. Kingdom of the Planet of the Apes	$5.5M	$158M	6	7.2 (53K)
4. The Garfield Movie	$4.8M	$78M	4	5.8 (8.4K)
5. IF	$3.6M	$101M	5	6.7 (14K)
6. The Watchers	$3.5M	$14M	2	5.8 (6.9K)
7. Furiosa: A Mad Max Saga	$2.6M	$63M	4	7.8 (85K)
8. The Fall Guy	$1.6M	$88M	7	7.0 (101K)
9. The Strangers: Chapter 1	$759K	$34M	5	4.7 (11K)
10. The Lord of the Rings: The Fellowship of the Ring	$633K	$319M	2	8.9 (2M)

Practice 2. From https://www.imdb.com/chart/moviemeter, prepare a table of most popular movies with movie name, year, length, and ratings.

In [7]:
```
#loading webpage content
webpage <- read_html("https://www.imdb.com/chart/moviemeter")
```

```
In [8]: movie <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]/div[2]") %>% html_text(trim=TRUE)
length(movie)
movie

year <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]/div[3]/span[1]") %>% html_text(trim=TRUE)
length(year)
year

length <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]/div[3]/span[2]") %>% html_text(trim=TRUE)
length(length)
length

grading <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]/div[3]/span[3]") %>% html_text(trim=TRUE)
length(grading)
grading

rating <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]/span/div/span[1]") %>% html_text(trim=TRUE)
length(rating)
rating
```

```
25
'Hit Man' · 'Bad Boys: Ride or Die' · 'Inside Out 2' · 'Furiosa: A Mad Max Saga' · 'Sous la Seine' · 'The Watchers' · 'The Fall Guy' · 'Gojira -1.0' · 'Civil War' · 'Inside Out' · 'Dune: Part Two' · 'Kingdom of the Planet of the Apes' · 'The Strangers: Chapter 1' · 'Munjya' · 'Challengers' · 'Deadpool & Wolverine' · 'The Bikeriders' · 'Mad Max: Fury Road' · 'Anyone But You' · 'The First Omen' · 'IF' · 'Am I OK?' · 'Kinds of Kindness' · 'The Ministry of Ungentlemanly Warfare' · 'Atlas'

25
'2023' · '2024' · '2024' · '2024' · '2024' · '2024' · '2024' · '2023' · '2024' · '2015' · '2024' · '2024' · '2024' · '2024' · '2024' · '2024' · '2023' · '2015' · '2023' · '2024' · '2024' · '2022' · '2024' · '2024' · '2024'

25
'1h 55m' · '1h 55m' · '1h 36m' · '2h 28m' · '1h 44m' · '1h 42m' · '2h 6m' · '2h 4m' · '1h 49m' · '1h 35m' · '2h 46m' · '2h 25m' · '1h 31m' · '2h 20m' · '2h 11m' · '2h 7m' · '1h 56m' · '2h' · '1h 43m' · '1h 59m' · '1h 44m' · '1h 26m' · '2h 44m' · '2h' · '1h 58m'

24
'R' · 'R' · 'PG' · 'R' · 'TV-MA' · 'PG-13' · 'PG-13' · 'PG-13' · 'R' · 'PG' · 'PG-13' · 'PG-13' · 'R' · 'R' · 'R' · 'R' · 'R' · 'R' · 'R' · 'PG' · 'R' · 'R' · 'R' · 'PG-13'

25
```

'7.0 (43K)' · '7.0 (19K)' · '8.0 (16K)' · '7.8 (85K)' · '5.2 (18K)' · '5.8 (6.9K)' · '7.0 (101K)' · '7.8 (119K)' · '7.2 (114K)' · '8.1 (795K)' · '8.6 (450K)' · '7.2 (53K)' · '4.7 (11K)' · '7.7 (12K)' · '7.3 (69K)' · '' · '7.4 (1.7K)' · '8.1 (1.1M)' · '6.1 (93K)' · '6.5 (35K)' · '6.7 (14K)' · '6.1 (4K)' · '6.9 (2.7K)' · '6.9 (44K)' · '5.6 (41K)'

In [9]:
```
df <- data.frame(movie, year, length, rating)
write.csv(df, file = 'practice2.csv', row.names=FALSE)
```

Web scraping using `for` loop

If you look at the above example, you will notice that grading is missing for a movie. In this case, creating a balanced table is not possible. To resolve this issue, we need to use looping.

In [10]:
```
elems <- webpage %>% html_nodes(xpath = "//div[contains(@class,'cli-children')]")

movie <- c()
year <- c()
length <- c()
rating <- c()
grading <- c()

for (e in elems) {
    val <- e %>% html_nodes(xpath = "div[2]") %>% html_text(trim=TRUE)
    movie <- c(movie, ifelse(length(val) == 0,"",val))

    val <- e %>% html_nodes(xpath = "div[3]/span[1]") %>% html_text(trim=TRUE)
    year <- c(year, ifelse(length(val) == 0,"",val))

    val <- e %>% html_nodes(xpath = "div[3]/span[2]") %>% html_text(trim=TRUE)
    length <- c(length, ifelse(length(val) == 0,"",val))

    val <- e %>% html_nodes(xpath = "div[3]/span[3]") %>% html_text(trim=TRUE)
    grading <- c(grading, ifelse(length(val) == 0,"",val))

    val <- e %>% html_nodes(xpath = "span/div/span[1]") %>% html_text(trim=TRUE)
    rating <- c(rating, ifelse(length(val) == 0,"",val))

}
```

In [11]:
```
df <- data.frame(movie, year, length, grading, rating)
head(df)
write.csv(df, file = 'practice2_forloop.csv', row.names=FALSE)
```

A data.frame: 6 × 5

	movie	year	length	grading	rating
	<chr>	<chr>	<chr>	<chr>	<chr>
1	Hit Man	2023	1h 55m	R	7.0 (43K)
2	Bad Boys: Ride or Die	2024	1h 55m	R	7.0 (19K)
3	Inside Out 2	2024	1h 36m	PG	8.0 (16K)
4	Furiosa: A Mad Max Saga	2024	2h 28m	R	7.8 (85K)
5	Sous la Seine	2024	1h 44m	TV-MA	5.2 (18K)
6	The Watchers	2024	1h 42m	PG-13	5.8 (6.9K)