Scraping top 250 movies on IMDB

Data Science in a Box
datasciencebox.org

Top 250 movies on IMDB

Top 250 movies on IMDB

Take a look at the source code, look for the tag `table` tag:
http://www.imdb.com/chart/top


```
599           <div class="desc">Showing <span>250</span> Titles</div>
600         </div>
601       </div>
602       <br class="clear">
603       <table class="chart full-width" data-caller-name="chart-top250movie">
604         <colgroup>
605           <col class="chartTableColumnPoster"/>
606           <col class="chartTableColumnTitle"/>
607           <col class="chartTableColumnIMDbRating"/>
608           <col class="chartTableColumnYourRating"/>
609           <col class="chartTableColumnWatchlistRibbon"/>
610         </colgroup>
611         <thead>
612         <tr>
613           <th></th>
614           <th>Rank &amp;amp; Title</th>
615           <th>IMDb Rating</th>
616           <th>Your Rating</th>
617           <th></th>
618         </tr>
619         </thead>
620         <tbody class="lister-list">
621
622   <tr>
623     <td class="posterColumn">
624
625     <span name="rk" data-value="1"></span>
626     <span name="ir" data-value="9.222796866017044"></span>
627     <span name="us" data-value="7.791552E11"></span>
628     <span name="nv" data-value="2297666"></span>
629     <span name="ur" data-value="-1.7772031339829564"></span>
630 <a href="/title/tt0111161/?pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=e31d89dd-322d-4646-8962-
    327b42fe94b1&pf_rd_r=RP41R6C3PS7J108DDRNN&pf_rd_s=center-
    1&pf_rd_t=15506&pf_rd_i=top&ref_=chttp_tt_1"
631 > <img src="https://m.media-
    amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRlYWMwMWFmXkEyXkFqcGde
    QXVyMTMxODk2OTU@._V1_UY67_CR0,0,45,67_AL_.jpg" width="45" height="67" alt="The
    Shawshank Redemption"/>
632 </a>     </td>
```

First check if you're allowed!

```
library(robotstxt)
paths_allowed("http://www.imdb.com")
```

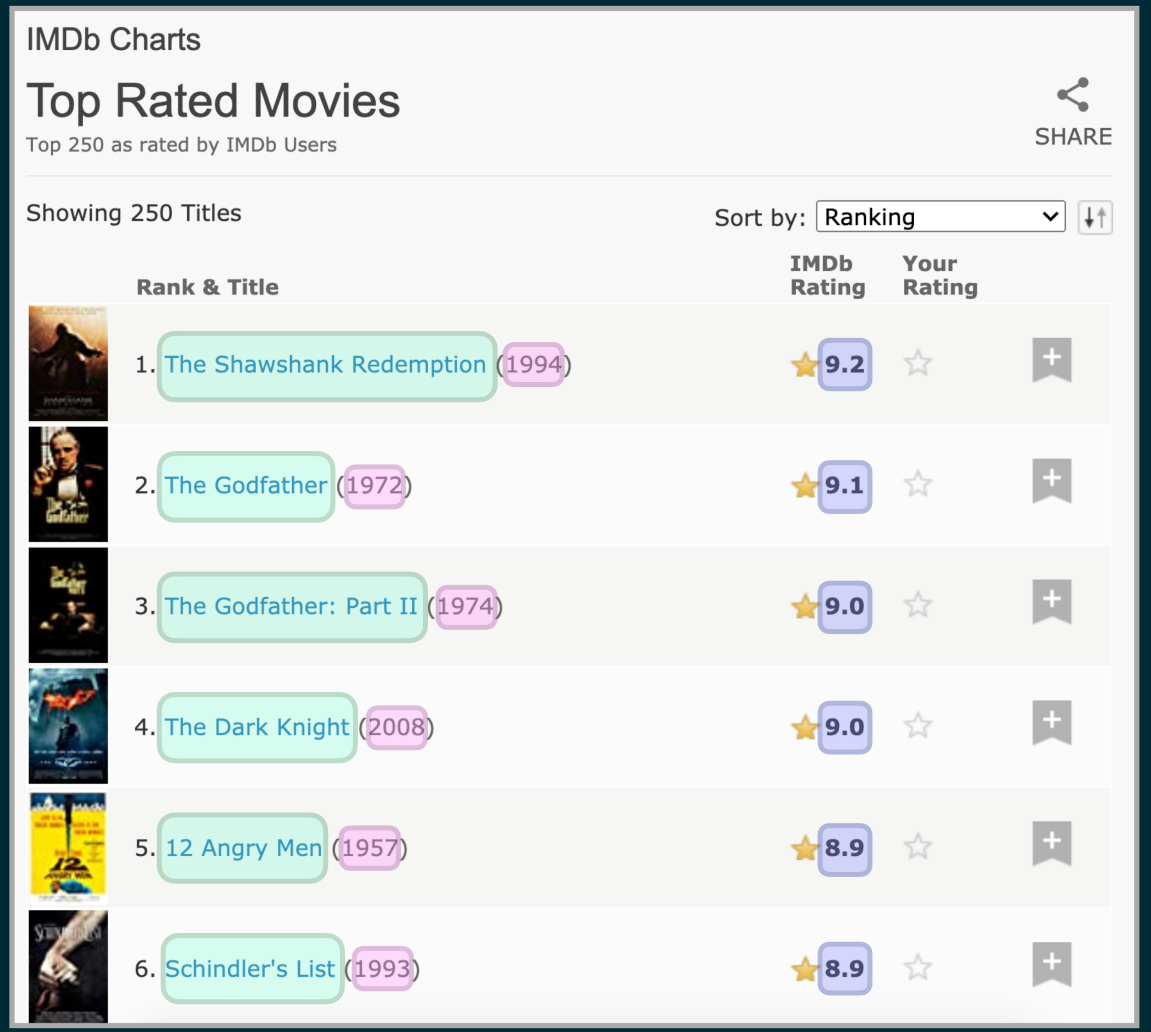
```
## [1] TRUE
```

vs. e.g.

```
paths_allowed("http://www.facebook.com")
```

```
## [1] FALSE
```

Plan



IMDb Charts

Top Rated Movies

Top 250 as rated by IMDb Users

SHARE

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. The Shawshank Redemption (1994)	9.2	
2. The Godfather (1972)	9.1	
3. The Godfather: Part II (1974)	9.0	
4. The Dark Knight (2008)	9.0	
5. 12 Angry Men (1957)	8.9	
6. Schindler's List (1993)	8.9	

imdb_top_250

title	year	rating

Plan

1. Read the whole page

2. Scrape movie titles and save as `titles`

3. Scrape years movies were made in and save as `years`

4. Scrape IMDB ratings and save as `ratings`

5. Create a data frame called `imdb_top_250` with variables `title`, `year`, and `rating`

Step 1. Read the whole page

Read the whole page

```
page <- read_html("https://www.imdb.com/chart/top/")
page
```

```
## {html_document}
## <html xmlns:og="http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html ...
## [2] <body id="styleguide-v2" class="fixed">\n            <img  ...
```

A webpage in R

- Result is a list with 2 elements

```
typeof(page)
```

```
## [1] "list"
```

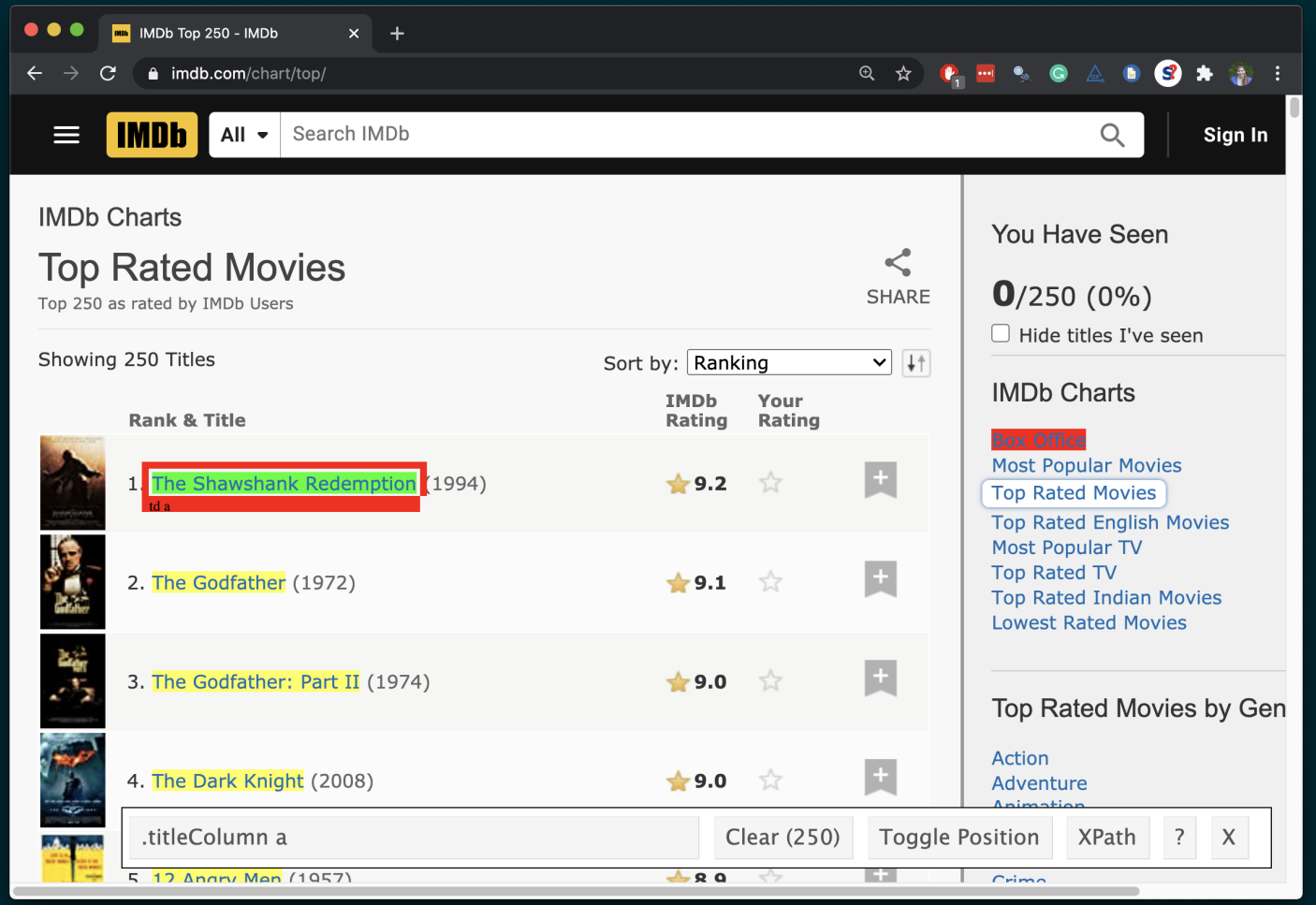
- that we need to convert to something more familiar, like a data frame....

```
class(page)
```

```
## [1] "xml_document" "xml_node"
```

Step 2. Scrape movie titles and save as `titles`

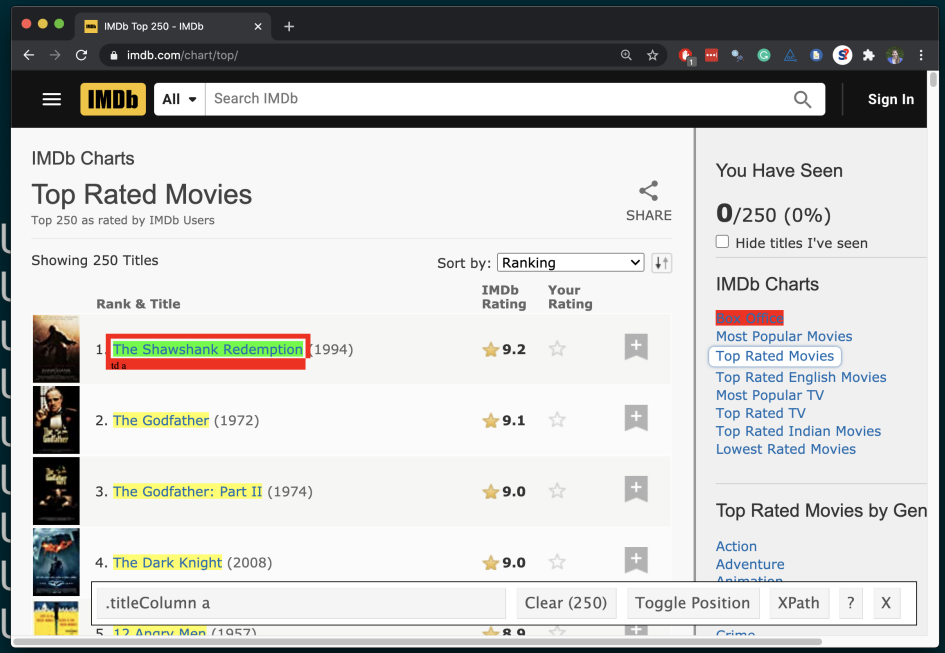
Scrape movie titles



Scrape the nodes

```
page %>%
  html_nodes(".titleColumn a")
```

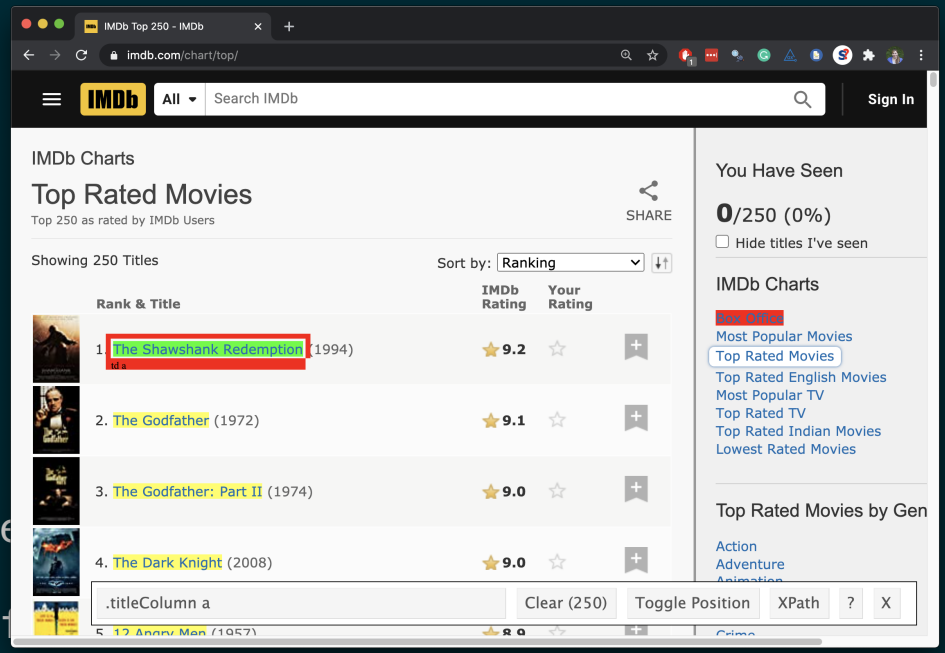
```
## {xml_nodeset (250)}
##  [1] <a href="/title/tt0111161/?pf_rd_m=A2FGELU
##  [2] <a href="/title/tt0068646/?pf_rd_m=A2FGELU
##  [3] <a href="/title/tt0468569/?pf_rd_m=A2FGELU
##  [4] <a href="/title/tt0071562/?pf_rd_m=A2FGELU
##  [5] <a href="/title/tt0050083/?pf_rd_m=A2FGELU
##  [6] <a href="/title/tt0108052/?pf_rd_m=A2FGELU
##  [7] <a href="/title/tt0167260/?pf_rd_m=A2FGELU
##  [8] <a href="/title/tt0110912/?pf_rd_m=A2FGELU
##  [9] <a href="/title/tt0120737/?pf_rd_m=A2FGELU
## [10] <a href="/title/tt0060196/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [11] <a href="/title/tt0109830/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [12] <a href="/title/tt0137523/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [13] <a href="/title/tt1375666/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [14] <a href="/title/tt0167261/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [15] <a href="/title/tt0080684/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
## [16] <a href="/title/tt0133093/?pf_rd_m=A2FGELUUNOQJNL&amp;pf_ ...
```



Extract the text from the nodes

```
page %>%
  html_nodes(".titleColumn a") %>%
  html_text()
```

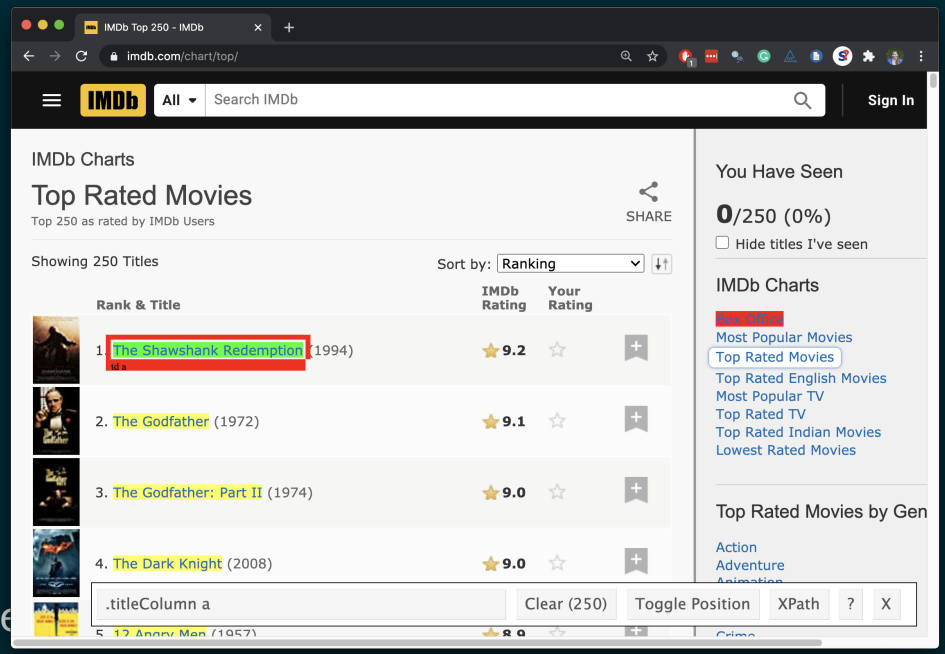
```
##   [1] "The Shawshank Redemption"
##   [2] "The Godfather"
##   [3] "The Dark Knight"
##   [4] "The Godfather: Part II"
##   [5] "12 Angry Men"
##   [6] "Schindler's List"
##   [7] "The Lord of the Rings: The Return of the
##   [8] "Pulp Fiction"
##   [9] "The Lord of the Rings: The Fellowship of
##  [10] "Il buono, il brutto, il cattivo"
##  [11] "Forrest Gump"
##  [12] "Fight Club"
##  [13] "Inception"
##  [14] "The Lord of the Rings: The Two Towers"
##  [15] "The Empire Strikes Back"
##  [16] "The Matrix"
```



Save as `titles`

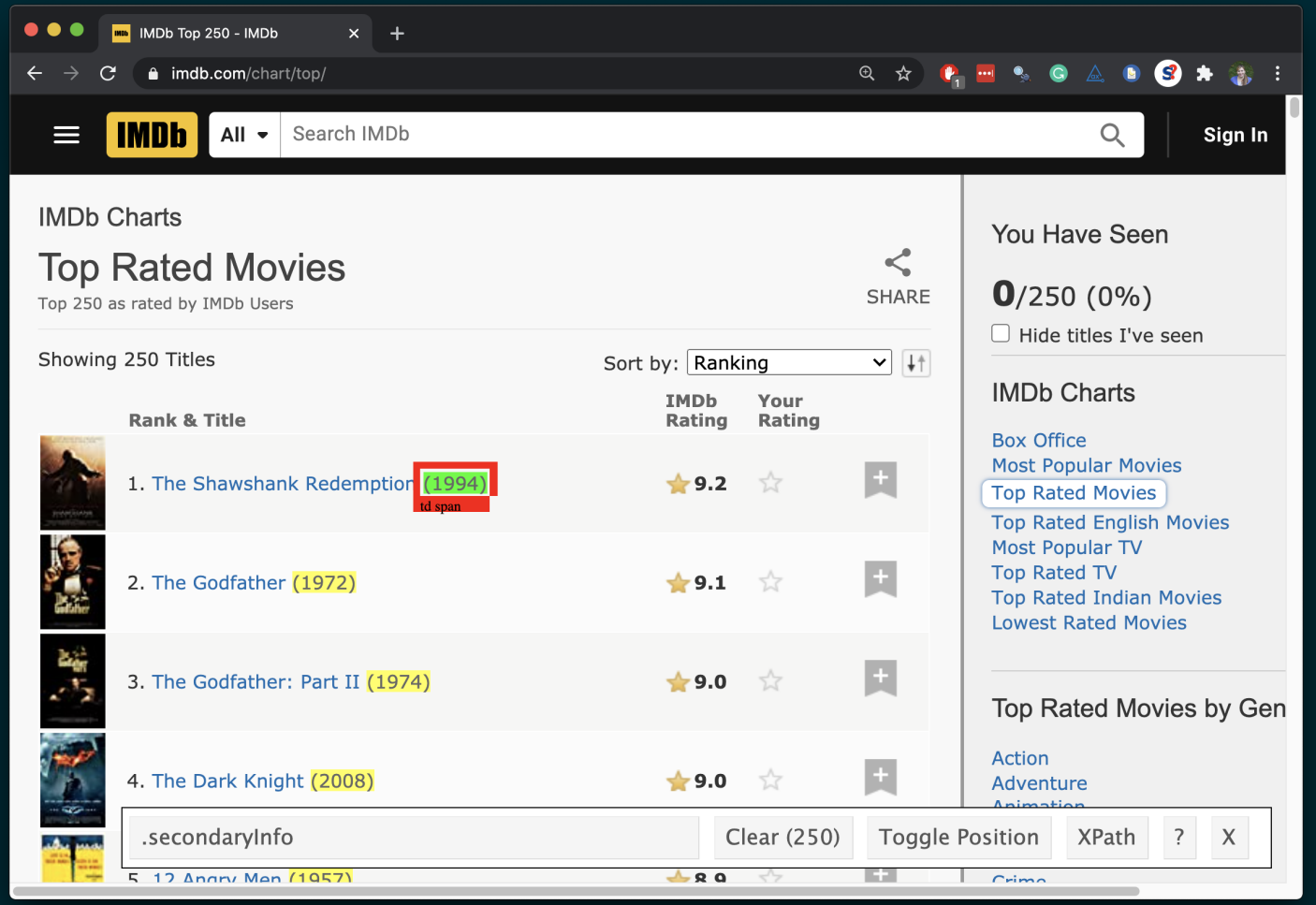
```
titles <- page %>%
  html_nodes(".titleColumn a") %>%
  html_text()

titles
```

```
##   [1] "The Shawshank Redemption"
##   [2] "The Godfather"
##   [3] "The Dark Knight"
##   [4] "The Godfather: Part II"
##   [5] "12 Angry Men"
##   [6] "Schindler's List"
##   [7] "The Lord of the Rings: The Return of the
##   [8] "Pulp Fiction"
##   [9] "The Lord of the Rings: The Fellowship of the Ring"
##  [10] "Il buono, il brutto, il cattivo"
##  [11] "Forrest Gump"
##  [12] "Fight Club"
##  [13] "Inception"
##  [14] "The Lord of the Rings: The Two Towers"
## ...
```



Step 3. Scrape year movies were made and save as `years`

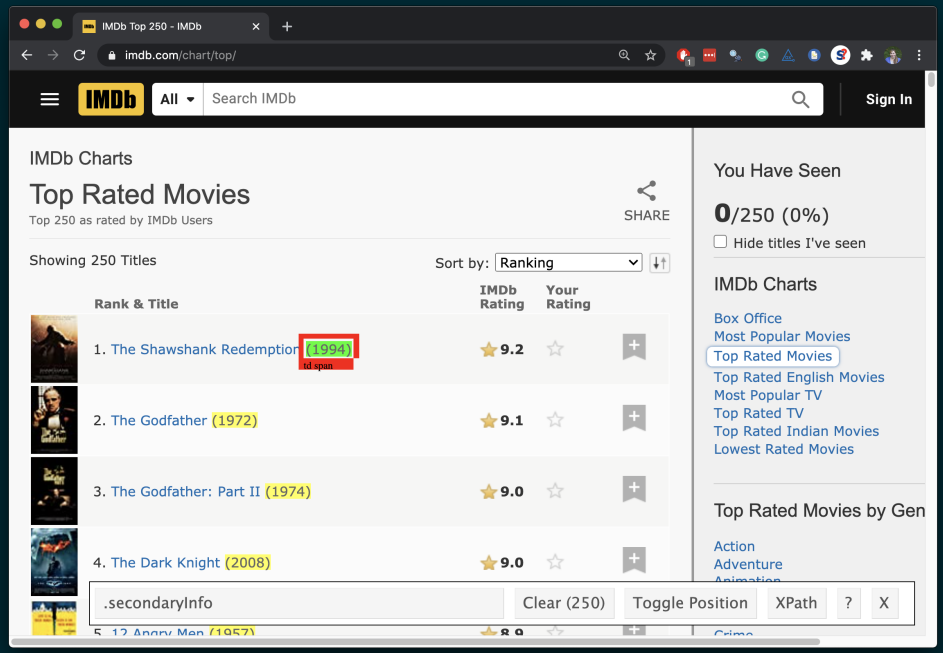
Scrape years movies were made in



Scrape the nodes

```
page %>%
  html_nodes(".secondaryInfo")
```

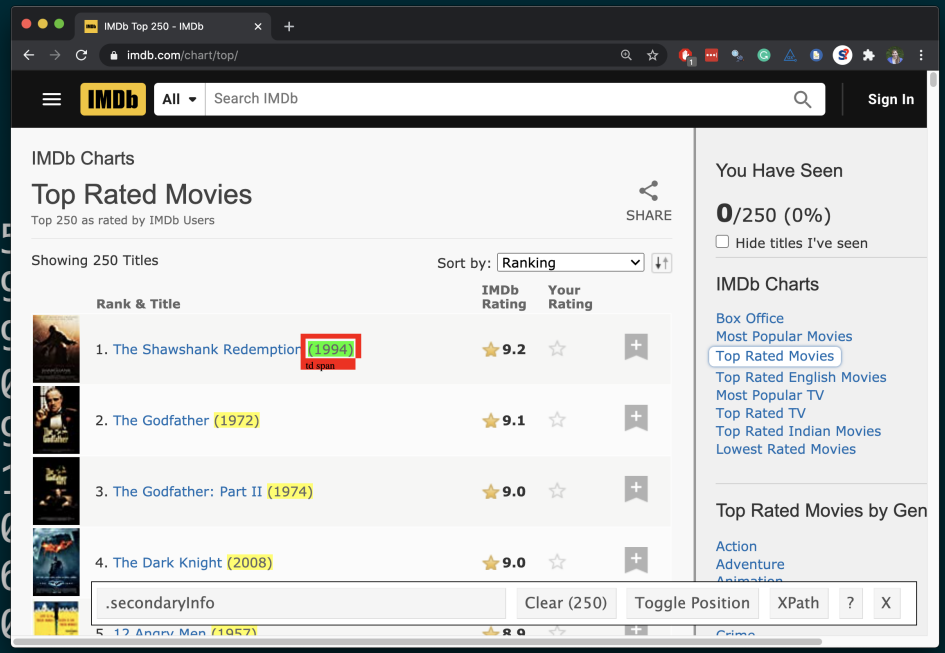
```
## {xml_nodeset (250)}
##  [1] <span class="secondaryInfo">(1994)</span>
##  [2] <span class="secondaryInfo">(1972)</span>
##  [3] <span class="secondaryInfo">(2008)</span>
##  [4] <span class="secondaryInfo">(1974)</span>
##  [5] <span class="secondaryInfo">(1957)</span>
##  [6] <span class="secondaryInfo">(1993)</span>
##  [7] <span class="secondaryInfo">(2003)</span>
##  [8] <span class="secondaryInfo">(1994)</span>
##  [9] <span class="secondaryInfo">(2001)</span>
## [10] <span class="secondaryInfo">(1966)</span>
## [11] <span class="secondaryInfo">(1994)</span>
## [12] <span class="secondaryInfo">(1999)</span>
## [13] <span class="secondaryInfo">(2010)</span>
## [14] <span class="secondaryInfo">(2002)</span>
## [15] <span class="secondaryInfo">(1980)</span>
## [16] <span class="secondaryInfo">(1999)</span>
```



Extract the text from the nodes

```
page %>%
  html_nodes(".secondaryInfo") %>%
  html_text()
```

```
##   [1] "(1994)" "(1972)" "(2008)" "(1974)" "(195
##   [7] "(2003)" "(1994)" "(2001)" "(1966)" "(199
##  [13] "(2010)" "(2002)" "(1980)" "(1999)" "(199
##  [19] "(1995)" "(1954)" "(1946)" "(1991)" "(200
##  [25] "(1997)" "(1999)" "(1977)" "(2014)" "(199
##  [31] "(2001)" "(1960)" "(2002)" "(1994)" "(201
##  [37] "(1998)" "(2000)" "(1995)" "(2006)" "(200
##  [43] "(2014)" "(2011)" "(1936)" "(1962)" "(196
##  [49] "(1954)" "(1979)" "(1931)" "(1988)" "(200
##  [55] "(1979)" "(1981)" "(2012)" "(2008)" "(2006)" "(1950)"
##  [61] "(1957)" "(1980)" "(1940)" "(1957)" "(2018)" "(1986)"
##  [67] "(1999)" "(1964)" "(2012)" "(2018)" "(2019)" "(2003)"
##  [73] "(1995)" "(1984)" "(1995)" "(2017)" "(1981)" "(2009)"
##  [79] "(1997)" "(1984)" "(2019)" "(1997)" "(2000)" "(2010)"
##  [85] "(1952)" "(2016)" "(1983)" "(2009)" "(1968)" "(1992)"
##  [91] "(2004)" "(1963)" "(1941)" "(2018)" "(1962)" "(1931)"
```



Clean up the text

We need to go from `"(1994)"` to `1994`:

- Remove `(` and `)`: string manipulation
- Convert to numeric: `as.numeric()`

stringr

- **stringr** provides a cohesive set of functions designed to make working with strings as easy as possible
- Functions in stringr start with `str_*()`, e.g.
  - `str_remove()` to remove a pattern from a string

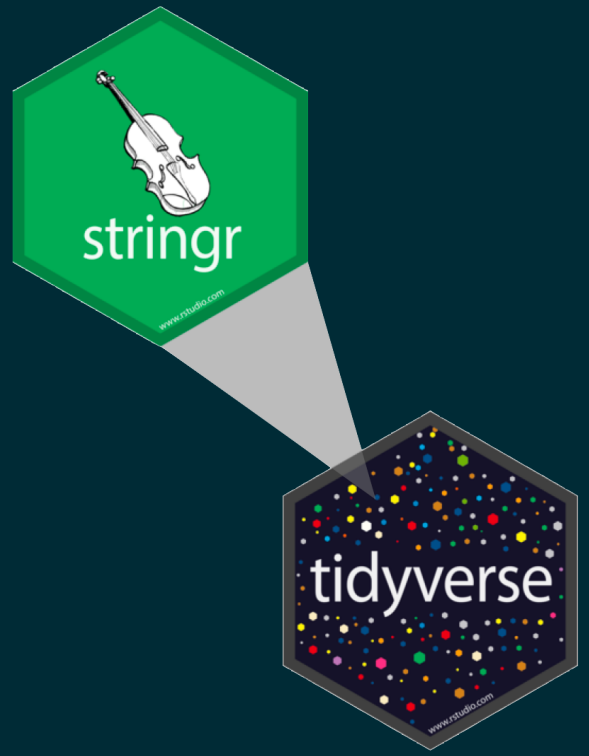
```
str_remove(string = "jello", pattern = "el")
```

```
## [1] "jlo"
```

  - `str_replace()` to replace a pattern with another

```
str_replace(string = "jello", pattern = "j", replacement =
```

```
## [1] "hello"
```



Clean up the text

```
page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") # remove (
```

```
##   [1] "1994)" "1972)" "2008)" "1974)" "1957)" "1993)" "2003)"
##   [8] "1994)" "2001)" "1966)" "1994)" "1999)" "2010)" "2002)"
##  [15] "1980)" "1999)" "1990)" "1975)" "1995)" "1954)" "1946)"
##  [22] "1991)" "2002)" "1998)" "1997)" "1999)" "1977)" "2014)"
##  [29] "1991)" "1985)" "2001)" "1960)" "2002)" "1994)" "2019)"
##  [36] "1994)" "1998)" "2000)" "1995)" "2006)" "2006)" "1942)"
##  [43] "2014)" "2011)" "1936)" "1962)" "1968)" "1988)" "1954)"
##  [50] "1979)" "1931)" "1988)" "2000)" "2022)" "1979)" "1981)"
##  [57] "2012)" "2008)" "2006)" "1950)" "1957)" "1980)" "1940)"
##  [64] "1957)" "2018)" "1986)" "1999)" "1964)" "2012)" "2018)"
##  [71] "2019)" "2003)" "1995)" "1984)" "1995)" "2017)" "1981)"
##  [78] "2009)" "1997)" "1984)" "2019)" "1997)" "2000)" "2010)"
##  [85] "1952)" "2016)" "1983)" "2009)" "1968)" "1992)" "2004)"
##  [92] "1963)" "1941)" "2018)" "1962)" "1931)" "2012)" "1959)"
##  [99] "1958)" "2001)" "1971)" "2021)" "1985)" "1987)" "1944)"
```

Clean up the text

```
page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") %>% # remove (
  str_remove("\\)") # remove )
```

```
##   [1] "1994" "1972" "2008" "1974" "1957" "1993" "2003" "1994"
##   [9] "2001" "1966" "1994" "1999" "2010" "2002" "1980" "1999"
##  [17] "1990" "1975" "1995" "1954" "1946" "1991" "2002" "1998"
##  [25] "1997" "1999" "1977" "2014" "1991" "1985" "2001" "1960"
##  [33] "2002" "1994" "2019" "1994" "1998" "2000" "1995" "2006"
##  [41] "2006" "1942" "2014" "2011" "1936" "1962" "1968" "1988"
##  [49] "1954" "1979" "1931" "1988" "2000" "2022" "1979" "1981"
##  [57] "2012" "2008" "2006" "1950" "1957" "1980" "1940" "1957"
##  [65] "2018" "1986" "1999" "1964" "2012" "2018" "2019" "2003"
##  [73] "1995" "1984" "1995" "2017" "1981" "2009" "1997" "1984"
##  [81] "2019" "1997" "2000" "2010" "1952" "2016" "1983" "2009"
##  [89] "1968" "1992" "2004" "1963" "1941" "2018" "1962" "1931"
##  [97] "2012" "1959" "1958" "2001" "1971" "2021" "1985" "1987"
## [105] "1944" "1960" "1983" "1976" "1962" "1952" "1973" "2020"
```

Convert to numeric

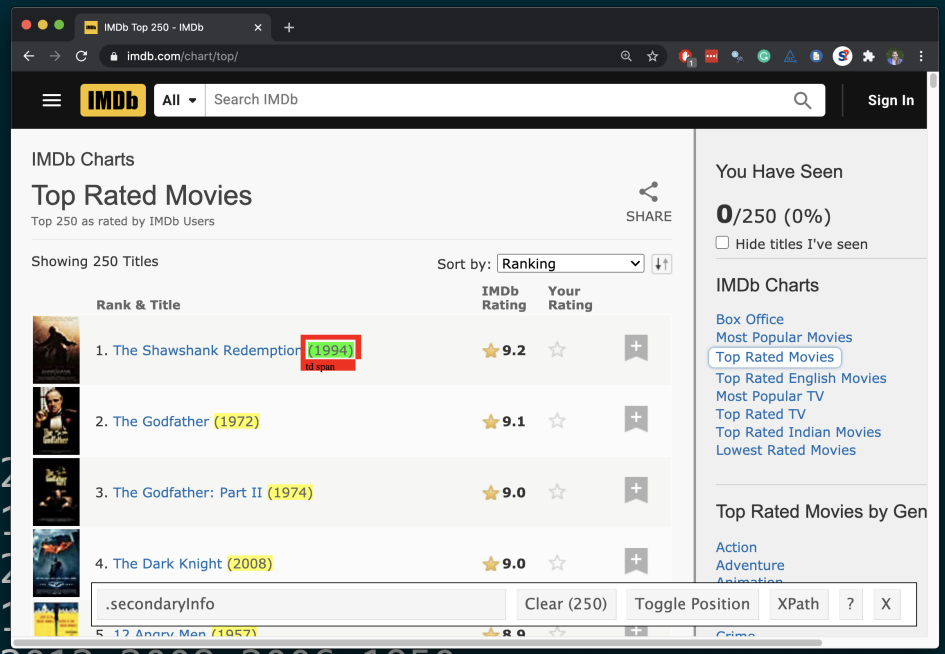
```
page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") %>% # remove (
  str_remove("\\)") %>% # remove )
  as.numeric()
```

```
##   [1] 1994 1972 2008 1974 1957 1993 2003 1994 2001 1966 1994 1999
##  [13] 2010 2002 1980 1999 1990 1975 1995 1954 1946 1991 2002 1998
##  [25] 1997 1999 1977 2014 1991 1985 2001 1960 2002 1994 2019 1994
##  [37] 1998 2000 1995 2006 2006 1942 2014 2011 1936 1962 1968 1988
##  [49] 1954 1979 1931 1988 2000 2022 1979 1981 2012 2008 2006 1950
##  [61] 1957 1980 1940 1957 2018 1986 1999 1964 2012 2018 2019 2003
##  [73] 1995 1984 1995 2017 1981 2009 1997 1984 2019 1997 2000 2010
##  [85] 1952 2016 1983 2009 1968 1992 2004 1963 1941 2018 1962 1931
##  [97] 2012 1959 1958 2001 1971 2021 1985 1987 1944 1960 1983 1976
## [109] 1962 1952 1973 2020 1997 2009 1995 1927 2000 2011 1988 2010
## [121] 1989 1948 2019 2007 2004 1965 2005 2016 1921 1959 2020 1950
## [133] 2018 2013 1961 1992 2006 1995 1985 2007 1975 2001 1999 1998
## [145] 1961 1948 1950 1963 2010 2003 1993 2007 1980 2003 1980 1974
```

Save as years

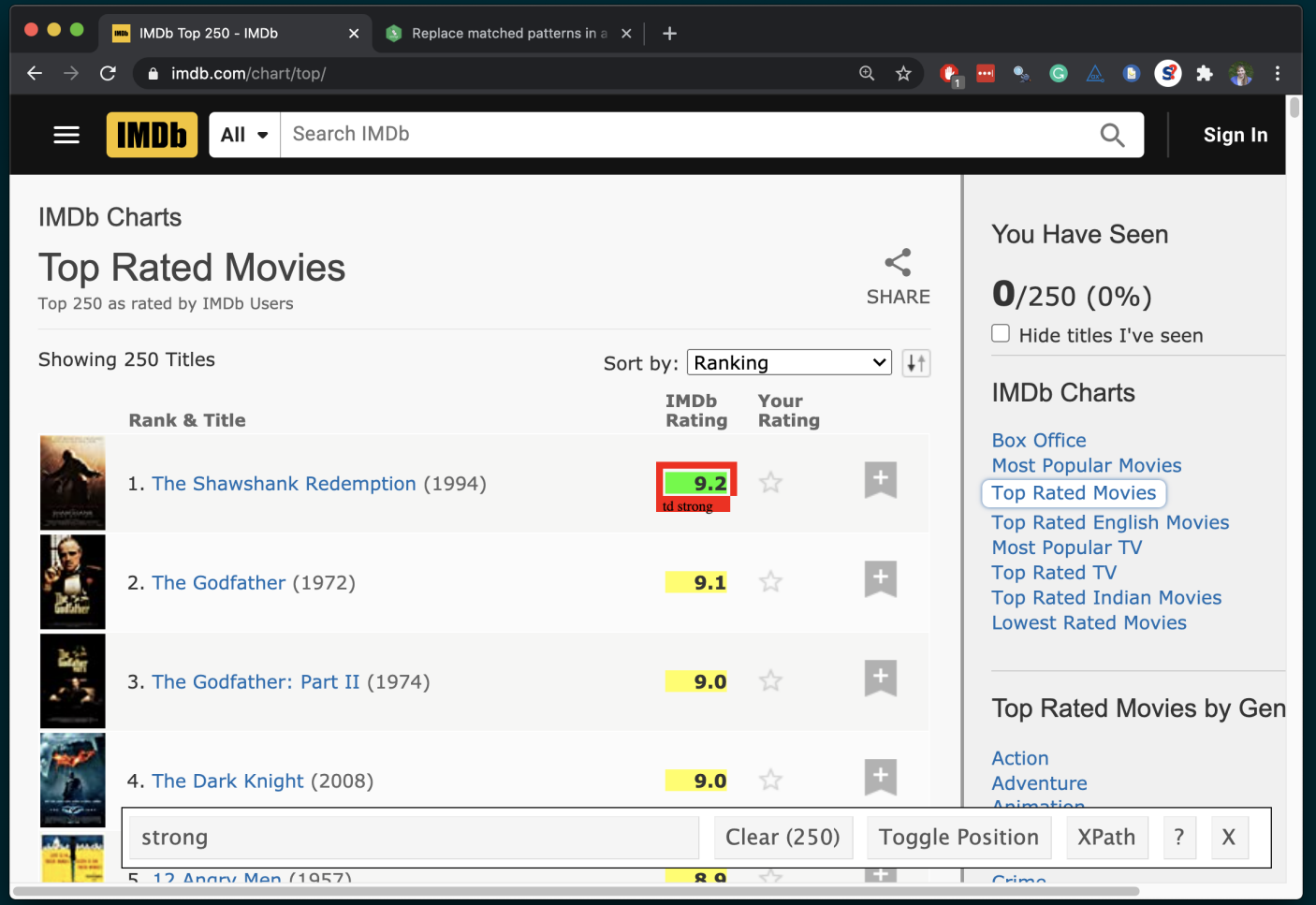
```
years <- page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") %>% # remove (
  str_remove("\\)") %>% # remove )
  as.numeric()

years
```

```
##   [1] 1994 1972 2008 1974 1957 1993 2003 1994 2
##  [13] 2010 2002 1980 1999 1990 1975 1995 1954 1
##  [25] 1997 1999 1977 2014 1991 1985 2001 1960 2
##  [37] 1998 2000 1995 2006 2006 1942 2014 2011 1
##  [49] 1954 1979 1931 1988 2000 2022 1979 1981 2012 2008 2006 1950
##  [61] 1957 1980 1940 1957 2018 1986 1999 1964 2012 2018 2019 2003
##  [73] 1995 1984 1995 2017 1981 2009 1997 1984 2019 1997 2000 2010
##  [85] 1952 2016 1983 2009 1968 1992 2004 1963 1941 2018 1962 1931
##  [97] 2012 1959 1958 2001 1971 2021 1985 1987 1944 1960 1983 1976
## [109] 1962 1952 1973 2020 1997 2009 1995 1927 2000 2011 1988 2010
## [121] 1989 1948 2019 2007 2004 1965 2005 2016 1921 1959 2020 1950
```



Step 4. Scrape IMDB ratings and save as `ratings`

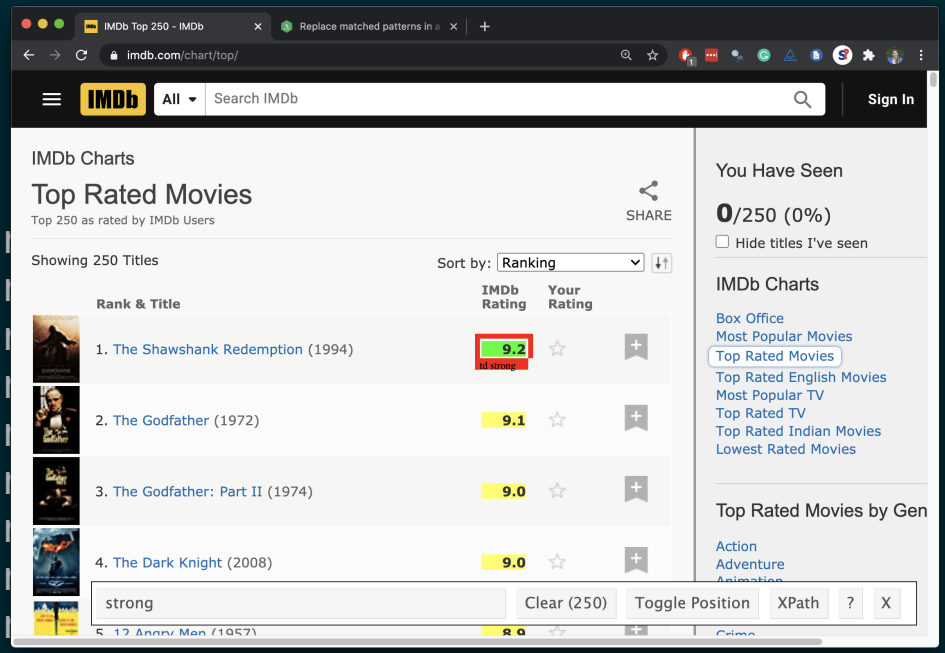
Scrape IMDB ratings



Scrape the nodes

```
page %>%
  html_nodes("strong")
```

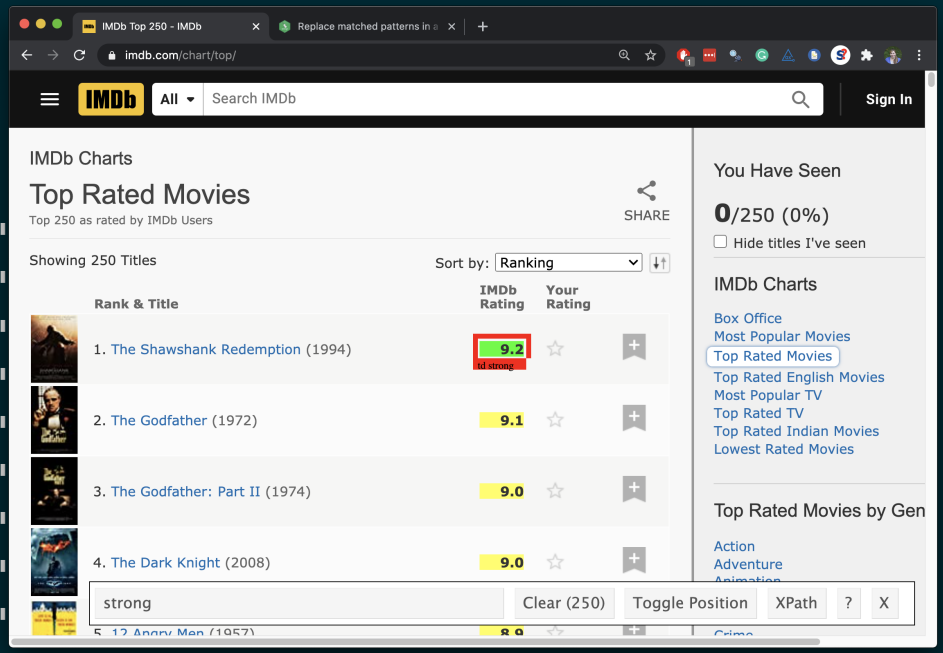
```
## {xml_nodeset (250)}
##  [1] <strong title="9.2 based on 2,583,442 user
##  [2] <strong title="9.2 based on 1,778,946 user
##  [3] <strong title="9.0 based on 2,553,377 user
##  [4] <strong title="9.0 based on 1,229,329 user
##  [5] <strong title="8.9 based on 763,028 user
##  [6] <strong title="8.9 based on 1,315,060 user
##  [7] <strong title="8.9 based on 1,775,663 user
##  [8] <strong title="8.9 based on 1,981,626 user
##  [9] <strong title="8.8 based on 1,797,008 user
## [10] <strong title="8.8 based on 741,812 user ratings">8.8</st ...
## [11] <strong title="8.8 based on 1,993,098 user ratings">8.8</ ...
## [12] <strong title="8.8 based on 2,033,560 user ratings">8.8</ ...
## [13] <strong title="8.7 based on 2,267,180 user ratings">8.7</ ...
## [14] <strong title="8.7 based on 1,603,769 user ratings">8.7</ ...
## [15] <strong title="8.7 based on 1,249,163 user ratings">8.7</ ...
## [16] <strong title="8.7 based on 1,857,023 user ratings">8.7</ ...
```



Extract the text from the nodes

```
page %>%
  html_nodes("strong") %>%
  html_text()
```

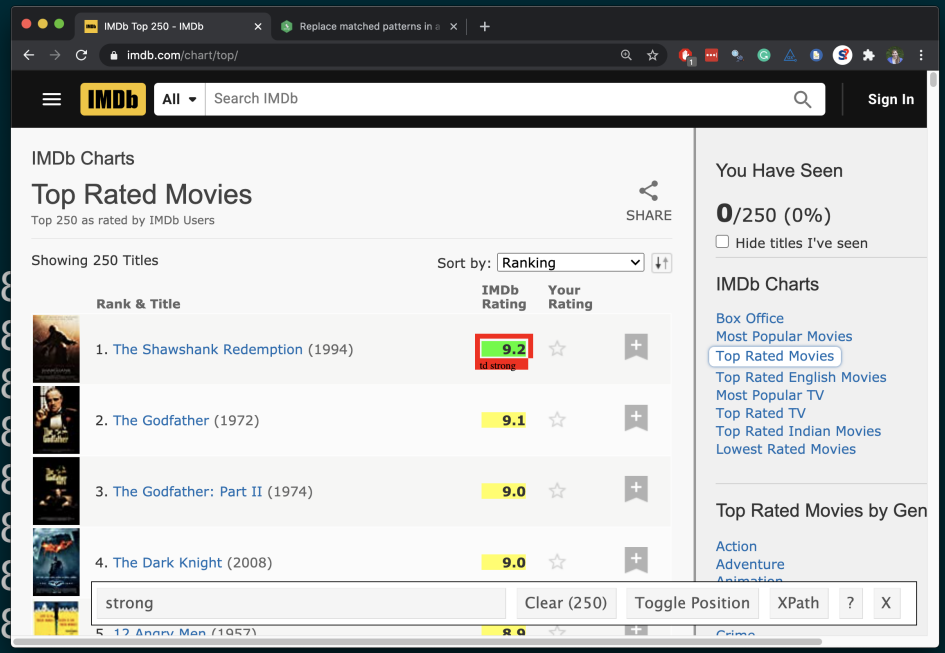
```
##   [1] "9.2" "9.2" "9.0" "9.0" "8.9" "8.9" "8.9"
##  [11] "8.8" "8.8" "8.7" "8.7" "8.7" "8.7" "8.7"
##  [21] "8.6" "8.6" "8.6" "8.6" "8.6" "8.6" "8.6"
##  [31] "8.5" "8.5" "8.5" "8.5" "8.5" "8.5" "8.5"
##  [41] "8.5" "8.5" "8.5" "8.5" "8.5" "8.4" "8.4"
##  [51] "8.4" "8.4" "8.4" "8.4" "8.4" "8.4" "8.4"
##  [61] "8.4" "8.4" "8.4" "8.4" "8.4" "8.3" "8.3"
##  [71] "8.3" "8.3" "8.3" "8.3" "8.3" "8.3" "8.3"
##  [81] "8.3" "8.3" "8.3" "8.3" "8.3" "8.3" "8.3"
##  [91] "8.3" "8.3" "8.3" "8.3" "8.3" "8.3" "8.3" "8.3" "8.2" "8.2"
## [101] "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2"
## [111] "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2"
## [121] "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2"
## [131] "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2" "8.2"
## [141] "8.2" "8.2" "8.2" "8.2" "8.1" "8.1" "8.1" "8.1" "8.1" "8.1"
## [151] "8.1" "8.1" "8.1" "8.1" "8.1" "8.1" "8.1" "8.1" "8.1" "8.1"
```



Convert to numeric

```
page %>%
  html_nodes("strong") %>%
  html_text() %>%
  as.numeric()
```

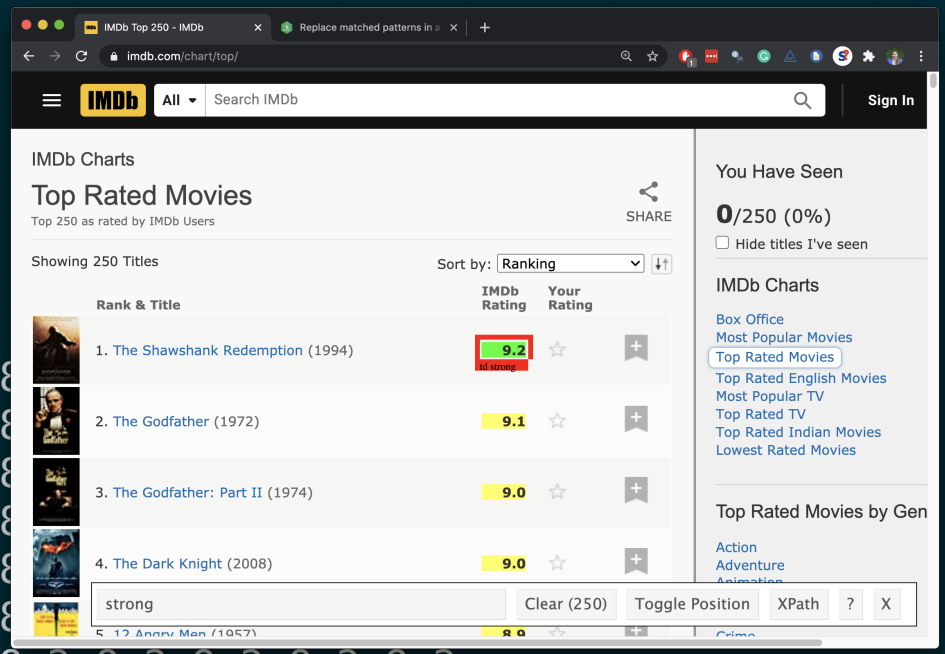
```
##   [1] 9.2 9.2 9.0 9.0 8.9 8.9 8.9 8.9 8.8 8.8 8
##  [16] 8.7 8.7 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8
##  [31] 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8
##  [46] 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8
##  [61] 8.4 8.4 8.4 8.4 8.4 8.3 8.3 8.3 8.3 8.3 8
##  [76] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8
##  [91] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.2 8.2 8
## [106] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8
## [121] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2
## [136] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.1 8.1 8.1 8.1 8.1 8.1
## [151] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
## [166] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
## [181] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
## [196] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
## [211] 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0
```



Save as ratings

```
ratings <- page %>%
  html_nodes("strong") %>%
  html_text() %>%
  as.numeric()

ratings
```

```
##   [1] 9.2 9.2 9.0 9.0 8.9 8.9 8.9 8.9 8.8 8.8 8
##  [16] 8.7 8.7 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8
##  [31] 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8
##  [46] 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8
##  [61] 8.4 8.4 8.4 8.4 8.4 8.3 8.3 8.3 8.3 8.3 8
##  [76] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8
##  [91] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.2 8.2 8.2 8.2 8.2 8.2 8.2
## [106] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2
## [121] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2
## [136] 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.2 8.1 8.1 8.1 8.1 8.1 8.1
## [151] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
## [166] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1
...
```



Step 5. Create a data frame called `imdb_top_250`

Create a data frame: imdb_top_250

```
imdb_top_250 <- tibble(
  title = titles,
  year = years,
  rating = ratings
  )

imdb_top_250
```

```
## # A tibble: 250 × 3
##   title                     year rating
##   <chr>                    <dbl>  <dbl>
## 1 The Shawshank Redemption  1994    9.2
## 2 The Godfather             1972    9.2
## 3 The Dark Knight           2008    9
## 4 The Godfather: Part II    1974    9
## 5 12 Angry Men              1957    8.9
## 6 Schindler's List          1993    8.9
## # … with 244 more rows
```

Show 10 entries Search:

	title	year	rating
1	The Shawshank Redemption	1994	9.2
2	The Godfather	1972	9.2
3	The Dark Knight	2008	9
4	The Godfather: Part II	1974	9
5	12 Angry Men	1957	8.9
6	Schindler's List	1993	8.9
7	The Lord of the Rings: The Return of the King	2003	8.9
8	Pulp Fiction	1994	8.9
9	The Lord of the Rings: The Fellowship of the Ring	2001	8.8
10	Il buono, il brutto, il cattivo	1966	8.8

Clean up / enhance

May or may not be a lot of work depending on how messy the data are

- See if you like what you got:

```
glimpse(imdb_top_250)
```

```
## Rows: 250
## Columns: 3
## $ title  <chr> "The Shawshank Redemption", "The Godfather", "Th…
## $ year   <dbl> 1994, 1972, 2008, 1974, 1957, 1993, 2003, 1994, …
## $ rating <dbl> 9.2, 9.2, 9.0, 9.0, 8.9, 8.9, 8.9, 8.9, 8.8, 8.8…
```

- Add a variable for rank

```
imdb_top_250 <- imdb_top_250 %>%
  mutate(rank = 1:nrow(imdb_top_250)) %>%
  relocate(rank)
```

```
## # A tibble: 250 × 4
##     rank title                                    year rating
##    <int> <chr>                                   <dbl>  <dbl>
##  1     1 The Shawshank Redemption                 1994    9.2
##  2     2 The Godfather                            1972    9.2
##  3     3 The Dark Knight                          2008    9
##  4     4 The Godfather: Part II                   1974    9
##  5     5 12 Angry Men                             1957    8.9
##  6     6 Schindler's List                         1993    8.9
##  7     7 The Lord of the Rings: The Return of the K…  2003    8.9
##  8     8 Pulp Fiction                             1994    8.9
##  9     9 The Lord of the Rings: The Fellowship of t…  2001    8.8
## 10    10 Il buono, il brutto, il cattivo          1966    8.8
## 11    11 Forrest Gump                             1994    8.8
## 12    12 Fight Club                               1999    8.8
## 13    13 Inception                                2010    8.7
## 14    14 The Lord of the Rings: The Two Towers    2002    8.7
## 15    15 The Empire Strikes Back                  1980    8.7
## 16    16 The Matrix                               1999    8.7
## 17    17 Goodfellas                               1990    8.7
## 18    18 One Flew Over the Cuckoo's Nest          1975    8.6
## 19    19 Se7en                                    1995    8.6
## 20    20 Shichinin no samurai                     1954    8.6
## # … with 230 more rows
```

What next?

Which years have the most movies on the list?

```
imdb_top_250 %>%
  count(year, sort = TRUE)
```

```
## # A tibble: 86 × 2
##    year     n
##   <dbl> <int>
## 1  1995     8
## 2  2004     7
## 3  1957     6
## 4  2003     6
## 5  2009     6
## 6  2019     6
## # … with 80 more rows
```

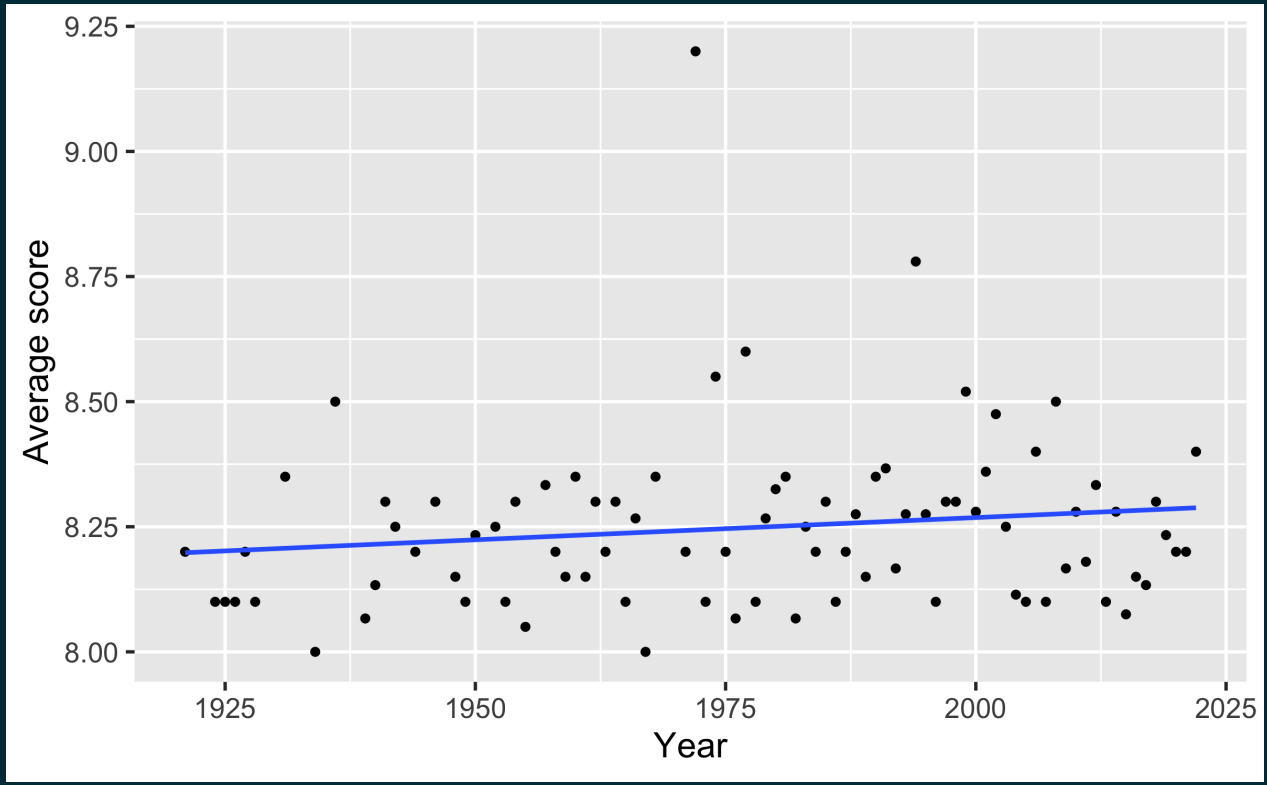
Which 1995 movies made the list?

```
imdb_top_250 %>%
  filter(year == 1995) %>%
  print(n = 8)
```

```
## # A tibble: 8 × 4
##    rank title                year rating
##   <int> <chr>               <dbl>  <dbl>
## 1    19 Se7en                1995    8.6
## 2    39 The Usual Suspects   1995    8.5
## 3    73 Braveheart           1995    8.3
## 4    75 Toy Story            1995    8.3
## 5   115 Heat                 1995    8.2
## 6   138 Casino               1995    8.2
## 7   186 Before Sunrise       1995    8.1
## 8   241 La haine             1995    8
```

Visualize the average yearly rating for movies that made it on the top 250 list over time.

Plot Code



Visualize the average yearly rating for movies that made it on the top 250 list over time.

Plot | Code

```
imdb_top_250 %>%
  group_by(year) %>%
  summarise(avg_score = mean(rating)) %>%
  ggplot(aes(y = avg_score, x = year)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Year", y = "Average score")
```