Web Scraping

craping top 250 movies on IMDB

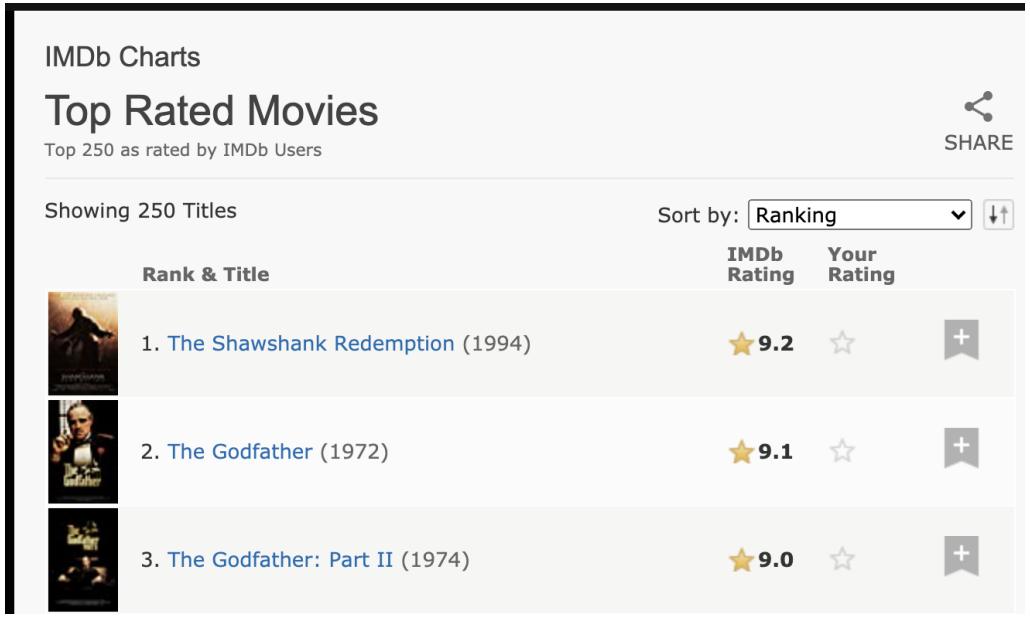
Prof. Dr. Jan Kirenz

The following content is based on Mine Çetinkaya-Rundel's excellent book Data Science in a Box

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")
```

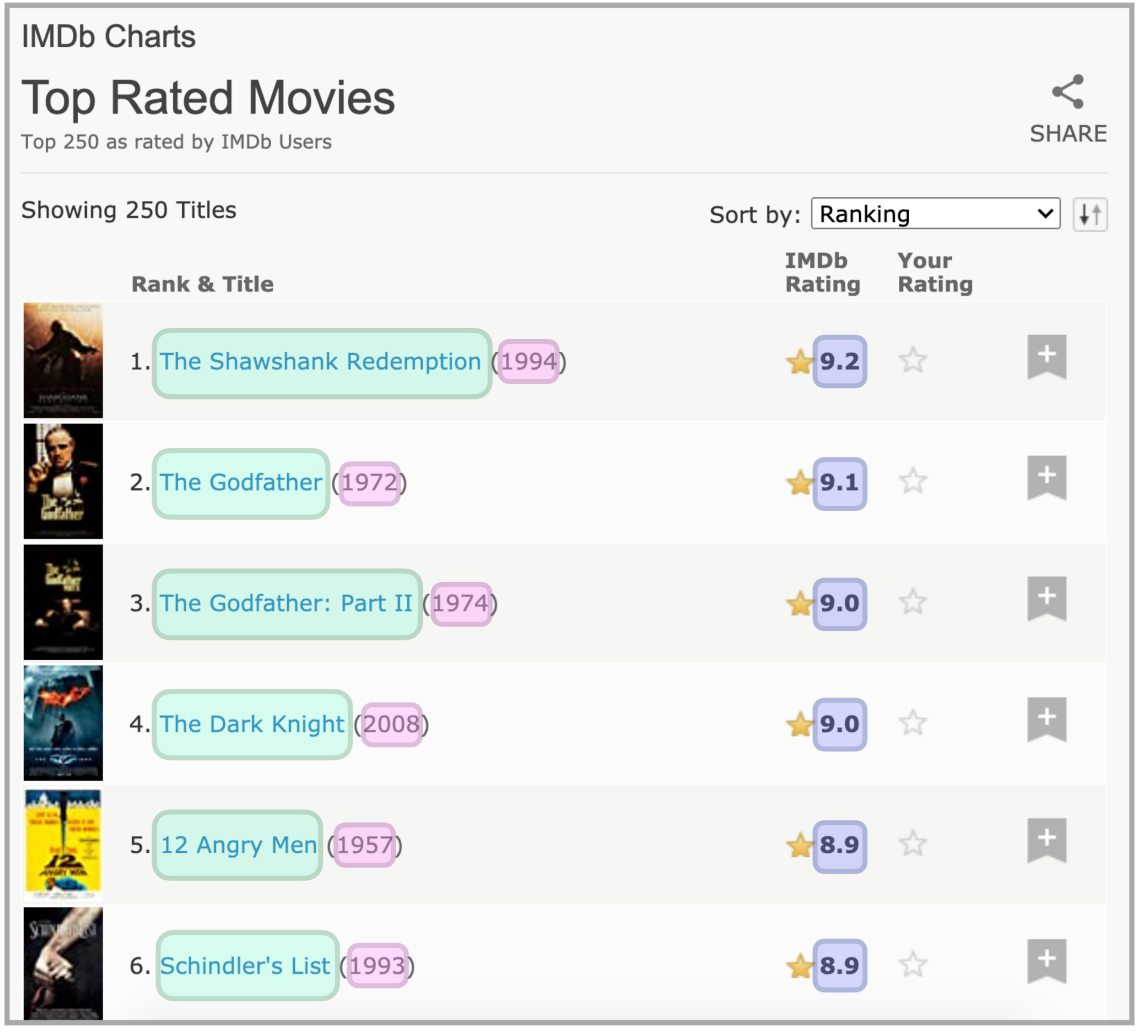
```
## [1] TRUE
```

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"
```

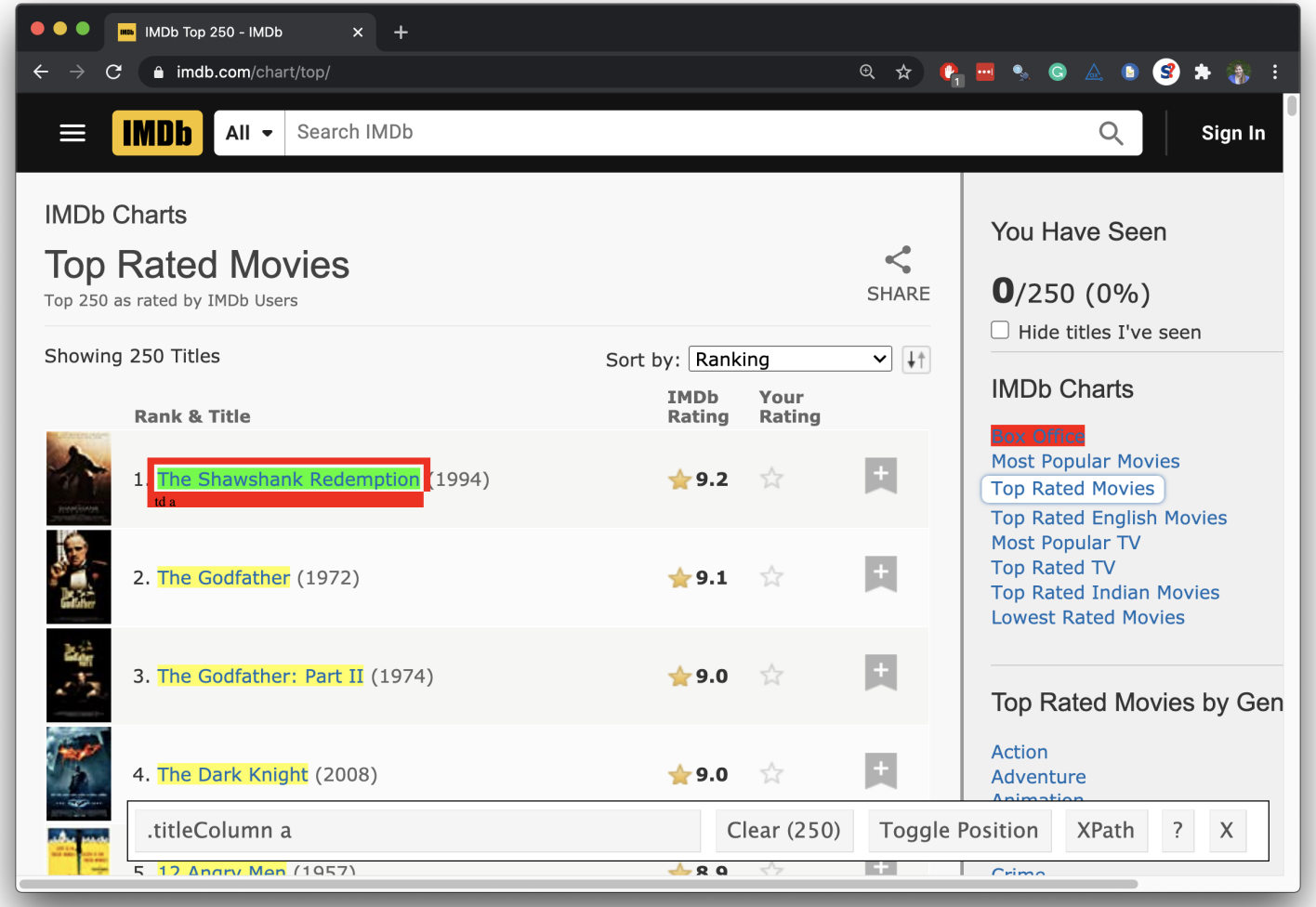
- that we need to convert to something more familiar, like a data frame....

```
class(page)
```

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

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

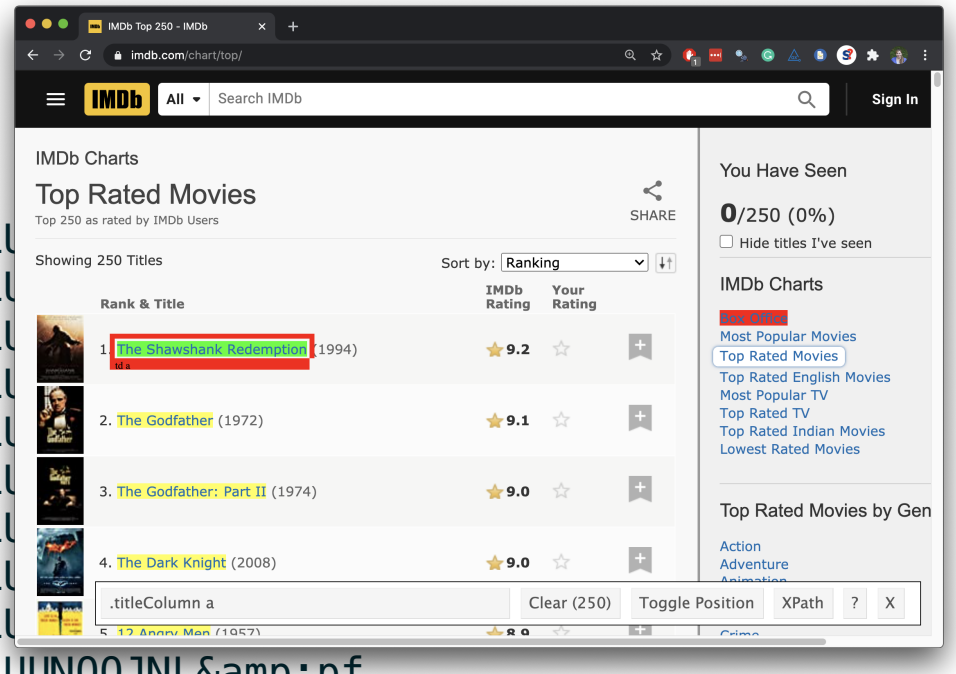
Scrape movie titles



Scrape the nodes

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

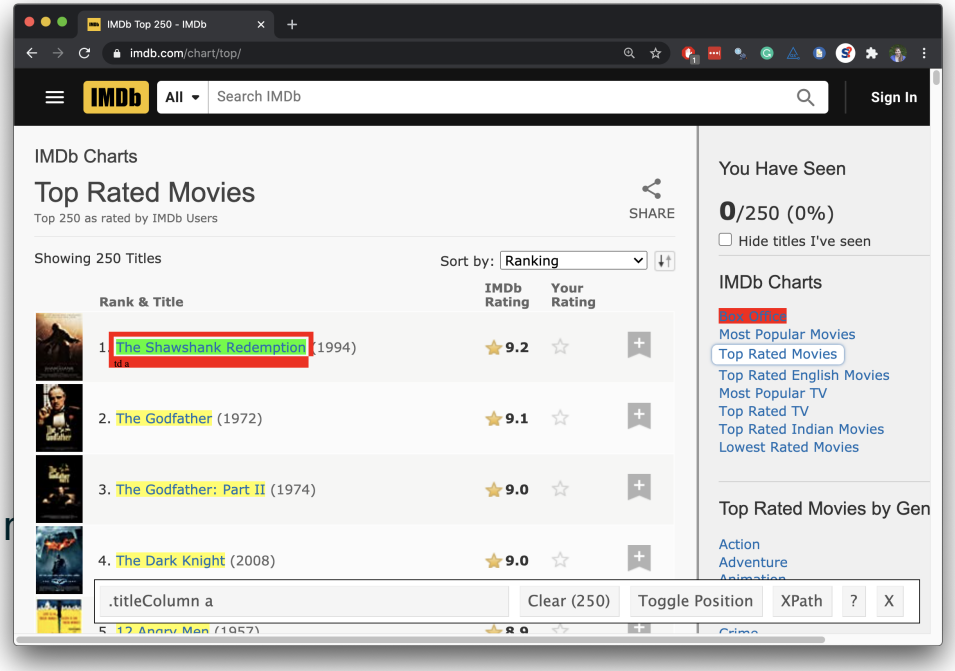
```
## {xml_nodeset (250)}
##   [1] <a href="/title/tt0111161/?pf_rd_m=A2FGELU
##   [2] <a href="/title/tt0068646/?pf_rd_m=A2FGELU
##   [3] <a href="/title/tt0071562/?pf_rd_m=A2FGELU
##   [4] <a href="/title/tt0468569/?pf_rd_m=A2FGELU
##   [5] <a href="/title/tt0050083/?pf_rd_m=A2FGELU
##   [6] <a href="/title/tt0108052/?pf_rd_m=A2FGELU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

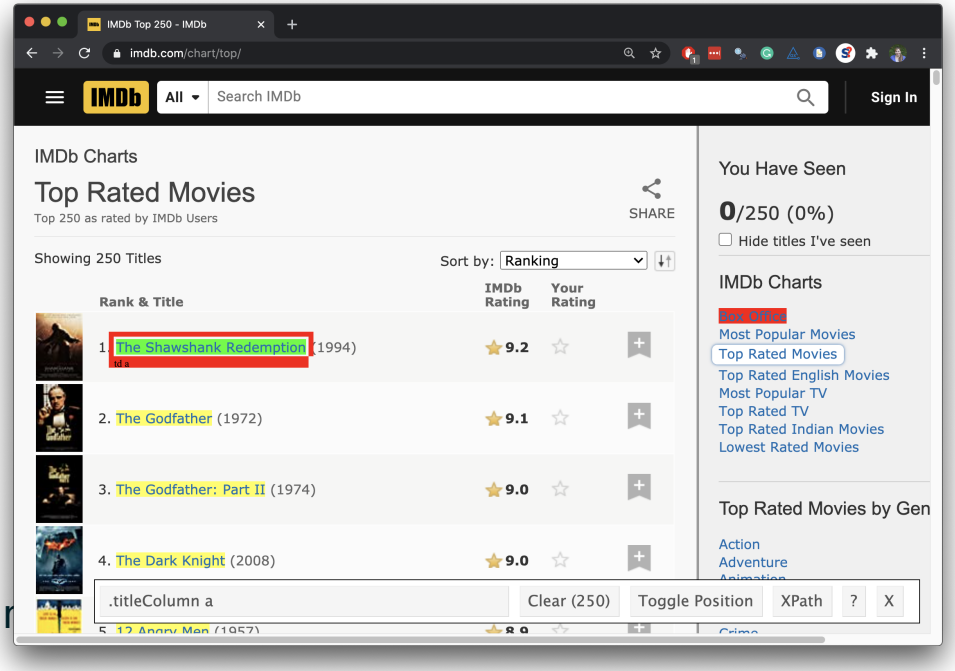
```
##    [1] "Die Verurteilten"
##    [2] "Der Pate"
##    [3] "Der Pate 2"
##    [4] "The Dark Knight"
##    [5] "Die zwölf Geschworenen"
##    [6] "Schindlers Liste"
##    [7] "Der Herr der Ringe: Die Rückkehr des Kön
##    [8] "Pulp Fiction"
##    [9] "Zwei glorreiche Halunken"
##   [10] "Der Herr der Ringe: Die Gefährten"
##   [11] "Fight Club"
##   [12] "Forrest Gump"
##   [13] "Inception"
##   [14] "Der Herr der Ringe: Die zwei Türme"
##   [15] "Das Imperium schlägt zurück"
##   [16] "Matrix"
...
```



Save as `titles`

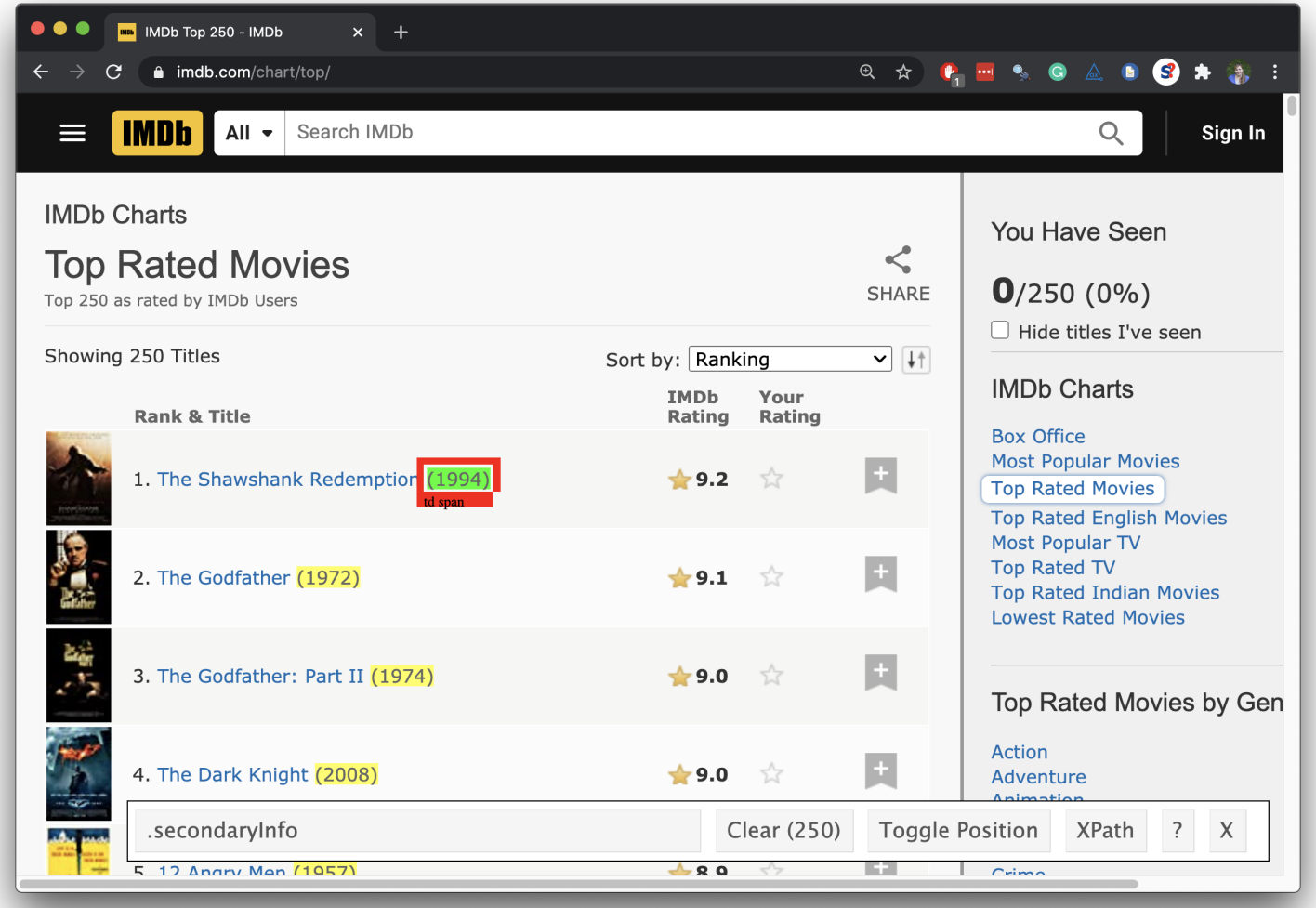
```
titles <- page %>%
  html_nodes(".titleColumn a") %>%
  html_text()

titles
```

```
##    [1] "Die Verurteilten"
##    [2] "Der Pate"
##    [3] "Der Pate 2"
##    [4] "The Dark Knight"
##    [5] "Die zwölf Geschworenen"
##    [6] "Schindlers Liste"
##    [7] "Der Herr der Ringe: Die Rückkehr des Kön
##    [8] "Pulp Fiction"
##    [9] "Zwei glorreiche Halunken"
##   [10] "Der Herr der Ringe: Die Gefährten"
##   [11] "Fight Club"
##   [12] "Forrest Gump"
##   [13] "Inception"
##   [14] "Der Herr der Ringe: Die zwei Türme"
...
```



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

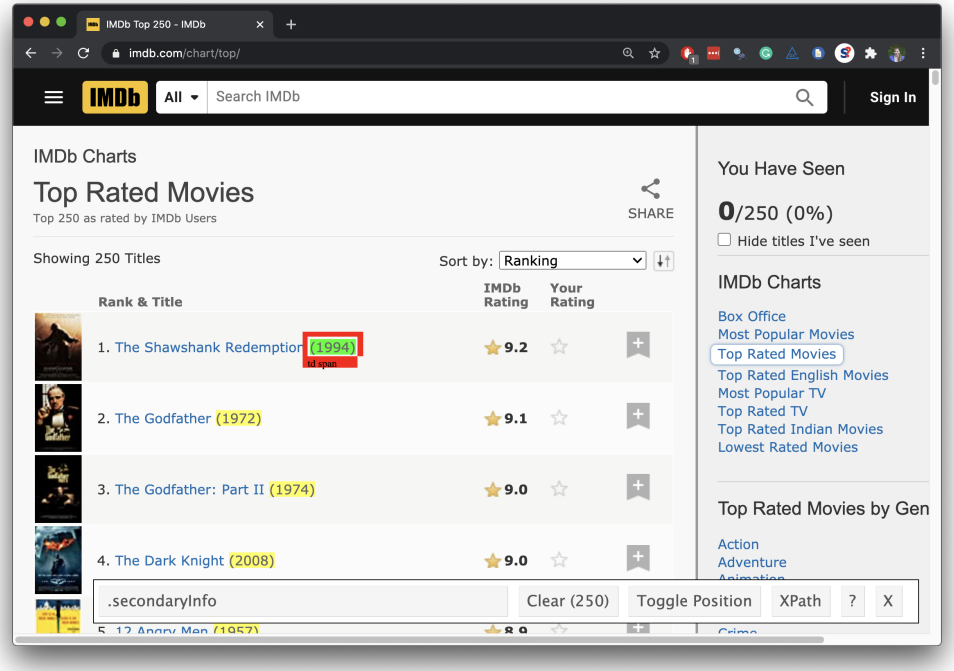
Scrape years movies were made in



Scrape the nodes

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

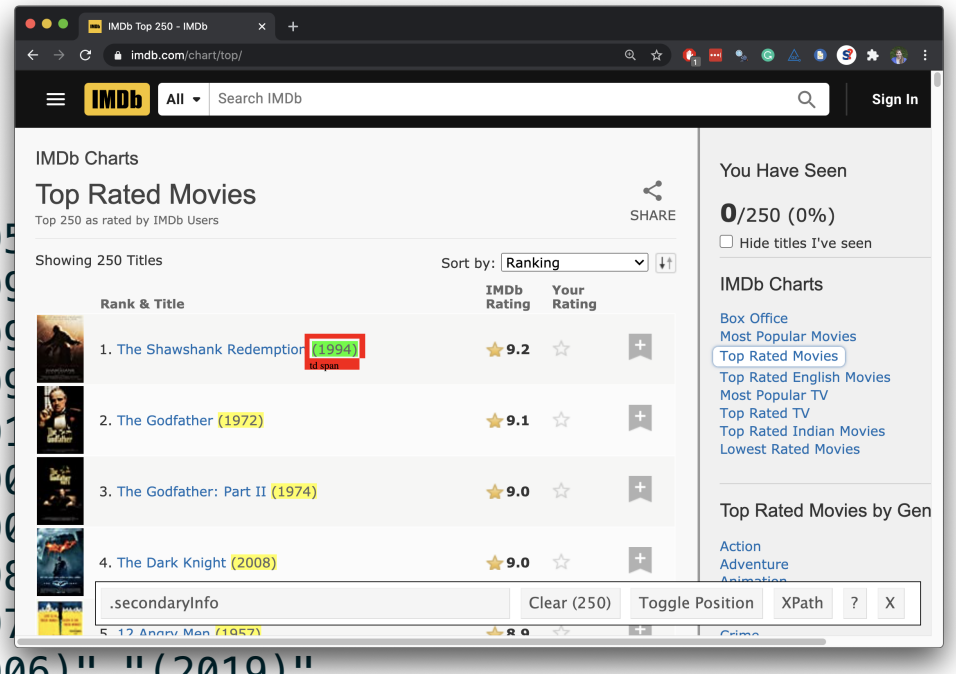
```
## {xml_nodeset (250)}
##  [1] <span class="secondaryInfo">(1994)</span>
##  [2] <span class="secondaryInfo">(1972)</span>
##  [3] <span class="secondaryInfo">(1974)</span>
##  [4] <span class="secondaryInfo">(2008)</span>
##  [5] <span class="secondaryInfo">(1957)</span>
##  [6] <span class="secondaryInfo">(1993)</span>
##  [7] <span class="secondaryInfo">(2003)</span>
##  [8] <span class="secondaryInfo">(1994)</span>
##  [9] <span class="secondaryInfo">(1966)</span>
## [10] <span class="secondaryInfo">(2001)</span>
## [11] <span class="secondaryInfo">(1999)</span>
## [12] <span class="secondaryInfo">(1994)</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)" "(1974)" "(2008)" "(195
##   [7] "(2003)" "(1994)" "(1966)" "(2001)" "(199
##  [13] "(2010)" "(2002)" "(1980)" "(1999)" "(199
##  [19] "(1954)" "(1995)" "(1997)" "(2002)" "(199
##  [25] "(1977)" "(1998)" "(2001)" "(1999)" "(201
##  [31] "(1994)" "(1995)" "(1962)" "(1994)" "(200
##  [37] "(1991)" "(1998)" "(1936)" "(1960)" "(200
##  [43] "(2006)" "(2011)" "(2014)" "(2006)" "(198
##  [49] "(1942)" "(1988)" "(1954)" "(2020)" "(197
##  [55] "(2000)" "(1940)" "(1981)" "(2012)" "(2006)" "(2019)"
##  [61] "(1957)" "(2008)" "(1980)" "(2018)" "(1950)" "(1957)"
##  [67] "(2003)" "(1997)" "(2018)" "(1964)" "(2012)" "(1984)"
##  [73] "(1986)" "(2016)" "(2017)" "(2019)" "(1999)" "(1995)"
##  [79] "(1981)" "(2009)" "(1995)" "(2018)" "(1963)" "(1984)"
##  [85] "(2009)" "(1983)" "(2007)" "(1997)" "(1992)" "(1968)"
##  [91] "(2000)" "(1958)" "(1931)" "(2012)" "(2004)" "(1941)"
...
```



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

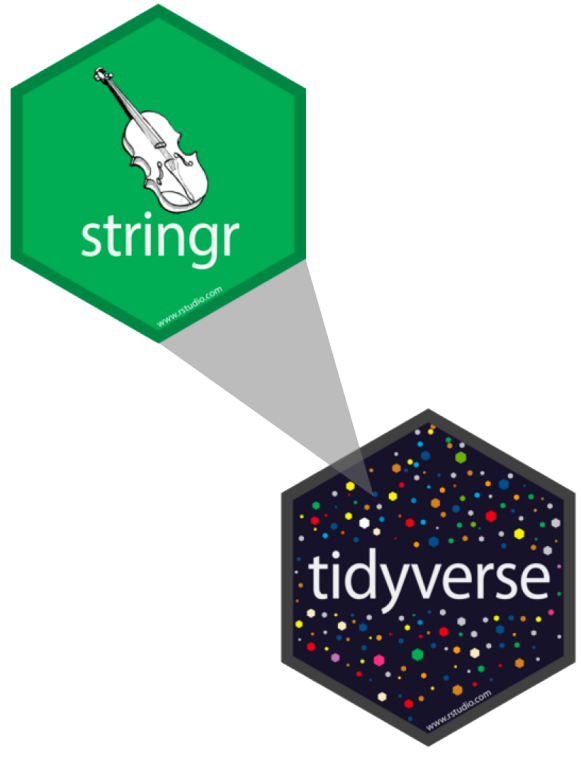
```
str_remove(string = "jello", pattern = "el")
```

```
## [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)" "1974)" "2008)" "1957)" "1993)" "2003)"
##   [8] "1994)" "1966)" "2001)" "1999)" "1994)" "2010)" "2002)"
##  [15] "1980)" "1999)" "1990)" "1975)" "1954)" "1995)" "1997)"
##  [22] "2002)" "1991)" "1946)" "1977)" "1998)" "2001)" "1999)"
##  [29] "2014)" "2019)" "1994)" "1995)" "1962)" "1994)" "2002)"
##  [36] "1985)" "1991)" "1998)" "1936)" "1960)" "2000)" "1931)"
##  [43] "2006)" "2011)" "2014)" "2006)" "1988)" "1968)" "1942)"
##  [50] "1988)" "1954)" "2020)" "1979)" "1979)" "2000)" "1940)"
##  [57] "1981)" "2012)" "2006)" "2019)" "1957)" "2008)" "1980)"
##  [64] "2018)" "1950)" "1957)" "2003)" "1997)" "2018)" "1964)"
##  [71] "2012)" "1984)" "1986)" "2016)" "2017)" "2019)" "1999)"
##  [78] "1995)" "1981)" "2009)" "1995)" "2018)" "1963)" "1984)"
##  [85] "2009)" "1983)" "2007)" "1997)" "1992)" "1968)" "2000)"
##  [92] "1958)" "1931)" "2012)" "2004)" "1941)" "2016)" "1987)"
##  [99] "1948)" "1952)" "1921)" "1959)" "2019)" "2000)" "1971)"
...
```

Clean up the text

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

```
##   [1] "1994" "1972" "1974" "2008" "1957" "1993" "2003" "1994"
##   [9] "1966" "2001" "1999" "1994" "2010" "2002" "1980" "1999"
##  [17] "1990" "1975" "1954" "1995" "1997" "2002" "1991" "1946"
##  [25] "1977" "1998" "2001" "1999" "2014" "2019" "1994" "1995"
##  [33] "1962" "1994" "2002" "1985" "1991" "1998" "1936" "1960"
##  [41] "2000" "1931" "2006" "2011" "2014" "2006" "1988" "1968"
##  [49] "1942" "1988" "1954" "2020" "1979" "1979" "2000" "1940"
##  [57] "1981" "2012" "2006" "2019" "1957" "2008" "1980" "2018"
##  [65] "1950" "1957" "2003" "1997" "2018" "1964" "2012" "1984"
##  [73] "1986" "2016" "2017" "2019" "1999" "1995" "1981" "2009"
##  [81] "1995" "2018" "1963" "1984" "2009" "1983" "2007" "1997"
##  [89] "1992" "1968" "2000" "1958" "1931" "2012" "2004" "1941"
##  [97] "2016" "1987" "1948" "1952" "1921" "1959" "2019" "2000"
## [105] "1971" "1983" "1952" "1976" "1985" "2010" "1962" "2001"
...
```

Convert to numeric

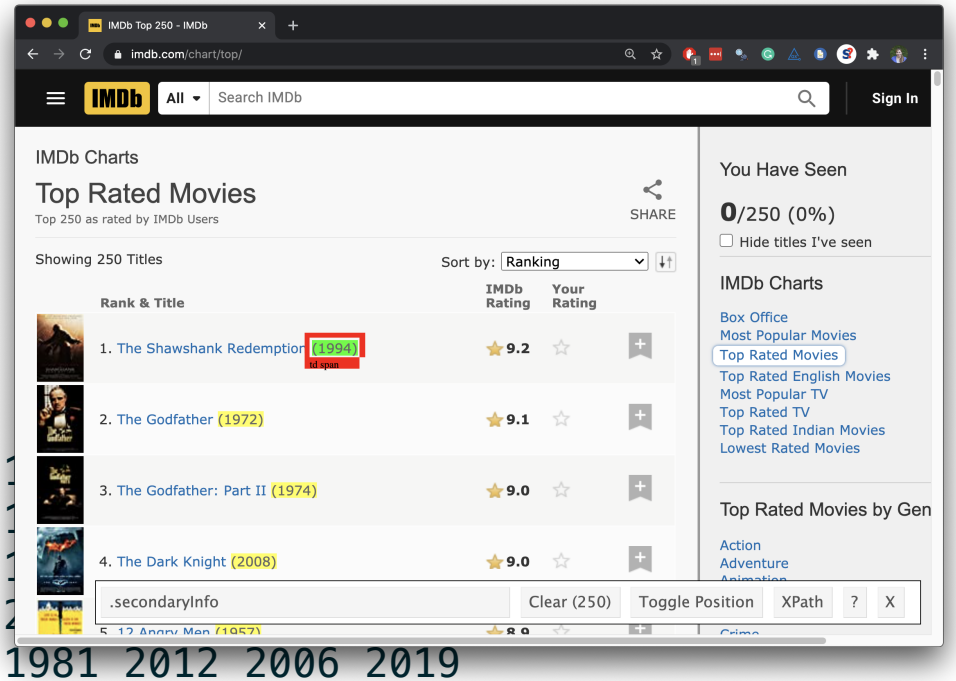
```
page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") %>% # remove (
  str_remove("\\)") %>% # remove )
  as.numeric()
```

```
##   [1] 1994 1972 1974 2008 1957 1993 2003 1994 1966 2001 1999 1994
##  [13] 2010 2002 1980 1999 1990 1975 1954 1995 1997 2002 1991 1946
##  [25] 1977 1998 2001 1999 2014 2019 1994 1995 1962 1994 2002 1985
##  [37] 1991 1998 1936 1960 2000 1931 2006 2011 2014 2006 1988 1968
##  [49] 1942 1988 1954 2020 1979 1979 2000 1940 1981 2012 2006 2019
##  [61] 1957 2008 1980 2018 1950 1957 2003 1997 2018 1964 2012 1984
##  [73] 1986 2016 2017 2019 1999 1995 1981 2009 1995 2018 1963 1984
##  [85] 2009 1983 2007 1997 1992 1968 2000 1958 1931 2012 2004 1941
##  [97] 2016 1987 1948 1952 1921 1959 2019 2000 1971 1983 1952 1976
## [109] 1985 2010 1962 2001 1973 2011 1927 2010 1965 1960 1944 1962
## [121] 2009 1989 1995 1997 1988 2018 1975 1961 2005 1950 2004 1997
## [133] 1992 1959 1985 2004 1950 2001 1995 1963 2013 2006 2009 2007
## [145] 1998 1988 1980 1961 1948 1954 2017 2010 1925 1974 2005 2007
...
```

Save as years

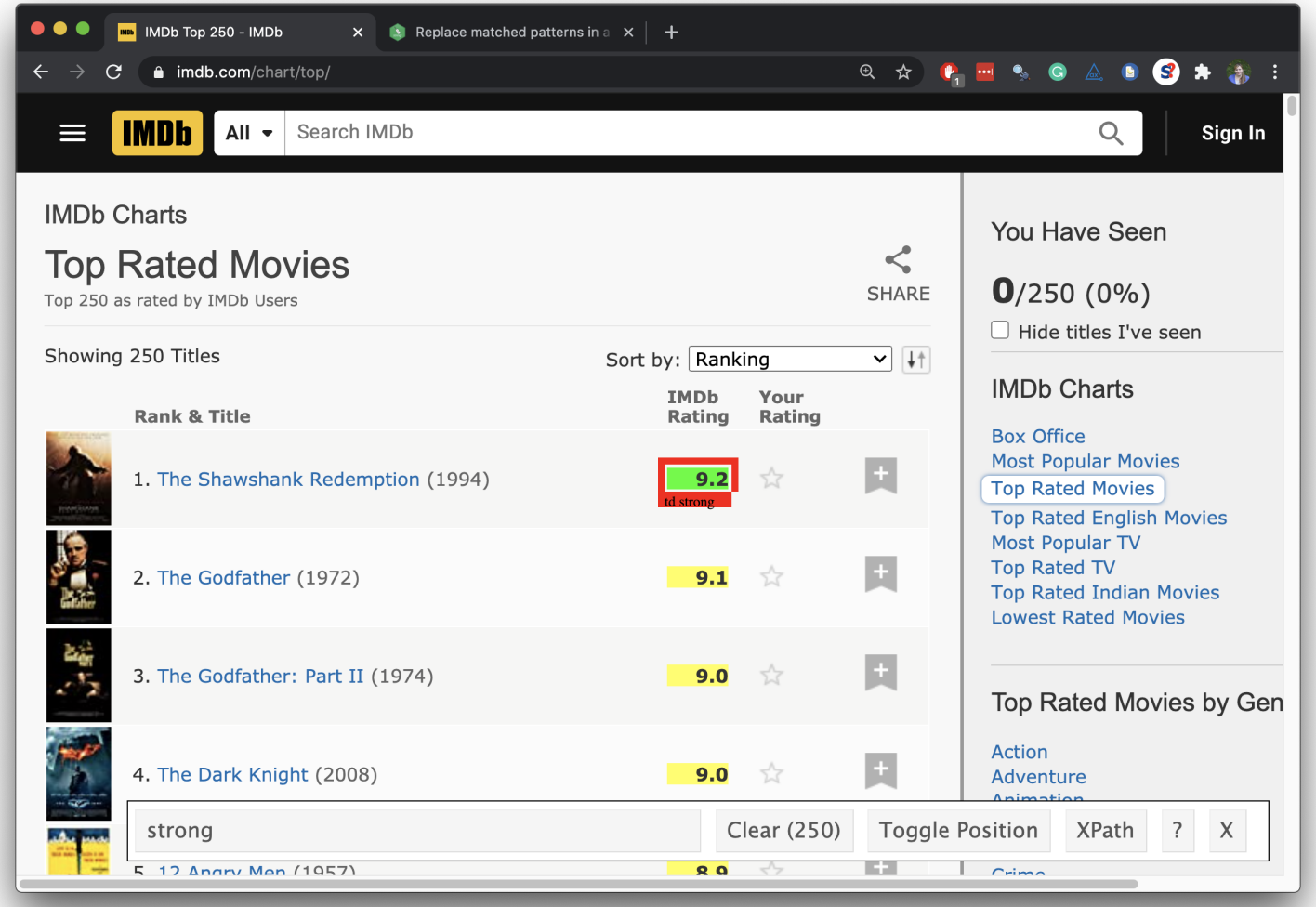
```
years <- page %>%
  html_nodes(".secondaryInfo") %>%
  html_text() %>%
  str_remove("\\(") %>% # remove (
  str_remove("\\)") %>% # remove )
  as.numeric()

years
```

```
##   [1] 1994 1972 1974 2008 1957 1993 2003 1994 1
##  [13] 2010 2002 1980 1999 1990 1975 1954 1995 1
##  [25] 1977 1998 2001 1999 2014 2019 1994 1995 1
##  [37] 1991 1998 1936 1960 2000 1931 2006 2011 2
##  [49] 1942 1988 1954 2020 1979 1979 2000 1940 1981 2012 2006 2019
##  [61] 1957 2008 1980 2018 1950 1957 2003 1997 2018 1964 2012 1984
##  [73] 1986 2016 2017 2019 1999 1995 1981 2009 1995 2018 1963 1984
##  [85] 2009 1983 2007 1997 1992 1968 2000 1958 1931 2012 2004 1941
##  [97] 2016 1987 1948 1952 1921 1959 2019 2000 1971 1983 1952 1976
## [109] 1985 2010 1962 2001 1973 2011 1927 2010 1965 1960 1944 1962
## [121] 2009 1989 1995 1997 1988 2018 1975 1961 2005 1950 2004 1997
...
```



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

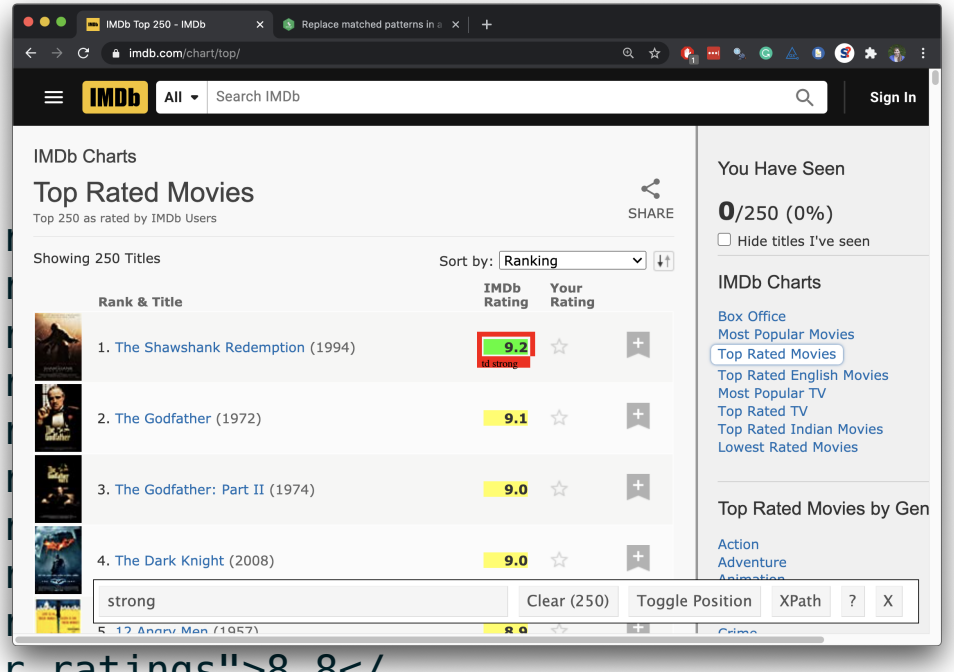
Scrape IMDB ratings



Scrape the nodes

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

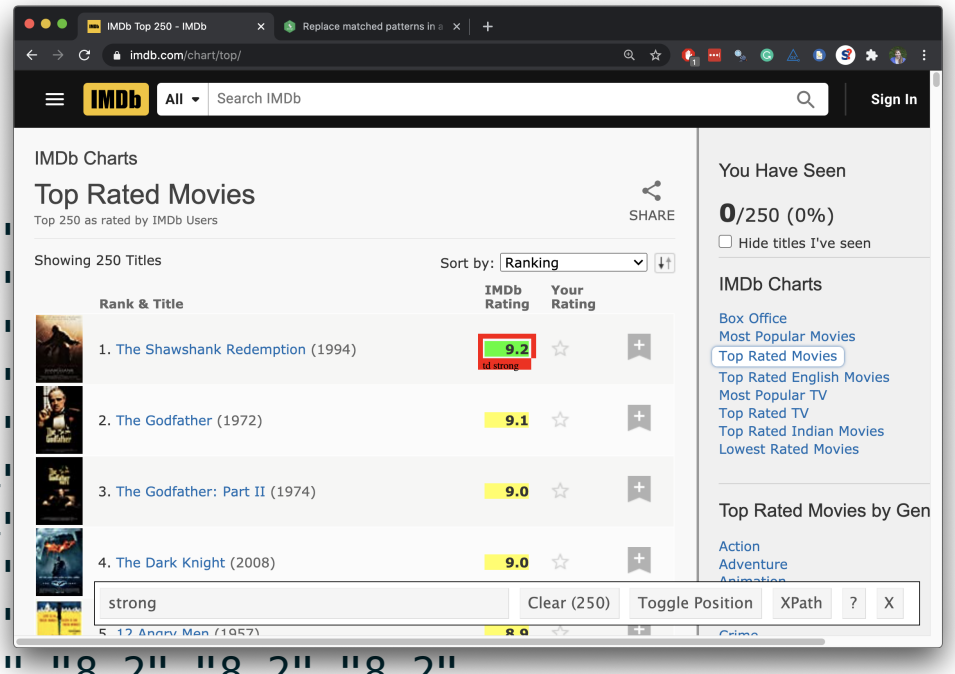
```
## {xml_nodeset (250)}
##  [1] <strong title="9.2 based on 2,329,241 user
##  [2] <strong title="9.1 based on 1,609,607 user
##  [3] <strong title="9.0 based on 1,123,632 user
##  [4] <strong title="9.0 based on 2,290,915 user
##  [5] <strong title="8.9 based on 685,620 user r
##  [6] <strong title="8.9 based on 1,207,495 user
##  [7] <strong title="8.9 based on 1,634,767 user
##  [8] <strong title="8.8 based on 1,816,667 user
##  [9] <strong title="8.8 based on 685,095 user r
## [10] <strong title="8.8 based on 1,651,799 user ratings">8.8</ ...
## [11] <strong title="8.8 based on 1,844,806 user ratings">8.8</ ...
## [12] <strong title="8.8 based on 1,797,892 user ratings">8.8</ ...
## [13] <strong title="8.7 based on 2,054,684 user ratings">8.7</ ...
## [14] <strong title="8.7 based on 1,478,142 user ratings">8.7</ ...
## [15] <strong title="8.7 based on 1,153,659 user ratings">8.7</ ...
## [16] <strong title="8.6 based on 1,667,186 user ratings">8.6</ ...
...
```



Extract the text from the nodes

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

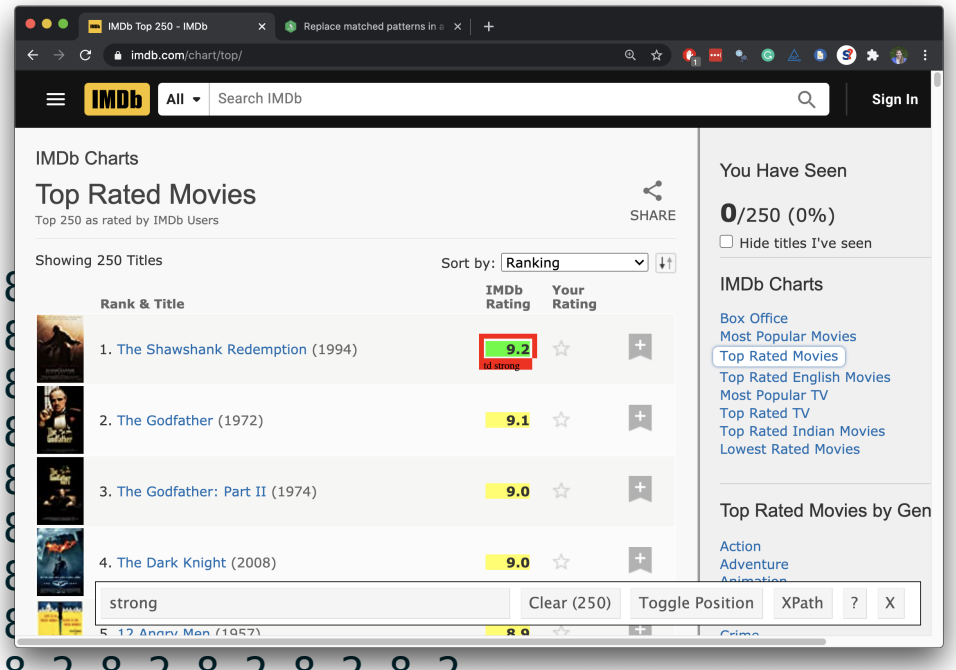
```
##   [1] "9.2" "9.1" "9.0" "9.0" "8.9" "8.9" "8.9"
##  [11] "8.8" "8.8" "8.7" "8.7" "8.7" "8.6" "8.6"
##  [21] "8.6" "8.6" "8.6" "8.6" "8.6" "8.5" "8.5"
##  [31] "8.5" "8.5" "8.5" "8.5" "8.5" "8.5" "8.5"
##  [41] "8.5" "8.5" "8.5" "8.5" "8.5" "8.5" "8.5"
##  [51] "8.4" "8.4" "8.4" "8.4" "8.4" "8.4" "8.4"
##  [61] "8.4" "8.4" "8.4" "8.4" "8.4" "8.4" "8.4"
##  [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" "8.2" "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" "8.1" "8.1" "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()
```

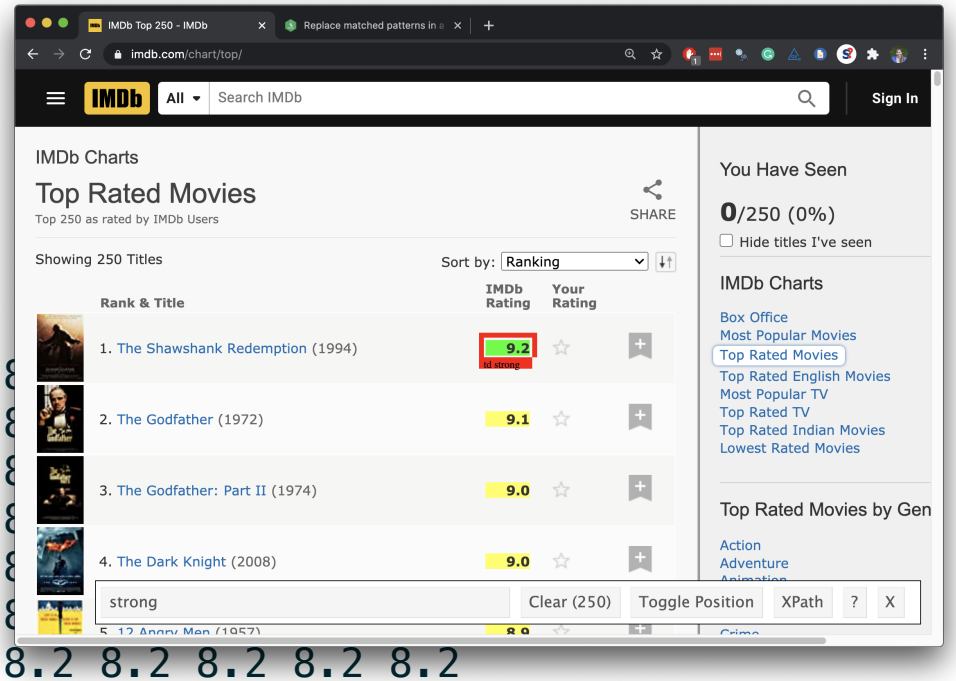
```
##   [1] 9.2 9.1 9.0 9.0 8.9 8.9 8.9 8.8 8.8 8.8 8
##  [16] 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8.6 8
##  [31] 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8
##  [46] 8.5 8.5 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8
##  [61] 8.4 8.4 8.4 8.4 8.4 8.4 8.4 8.3 8.3 8.3 8
##  [76] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8
##  [91] 8.3 8.3 8.3 8.3 8.3 8.3 8.3 8.2 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 8.1 8.1 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] 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 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 9.1 9.0 9.0 8.9 8.9 8.9 8.8 8.8 8.8 8
##  [16] 8.6 8.6 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] 8.5 8.5 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 8.4 8.4 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 8.2 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 8.1 8.1 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 x 3
##   title                    year rating
##   <chr>                   <dbl>  <dbl>
## 1 Die Verurteilten         1994    9.2
## 2 Der Pate                 1972    9.1
## 3 Der Pate 2               1974    9
## 4 The Dark Knight          2008    9
## 5 Die zwölf Geschworenen   1957    8.9
## 6 Schindlers Liste         1993    8.9
## # … with 244 more rows
```

Show 10 entries Search:

1	Die Verurteilten	1994	9.2
2	Der Pate	1972	9.1
3	Der Pate 2	1974	9
4	The Dark Knight	2008	9
5	Die zwölf Geschworenen	1957	8.9
6	Schindlers Liste	1993	8.9
7	Der Herr der Ringe: Die Rückkehr des Königs	2003	8.9
8	Pulp Fiction	1994	8.8
9	Zwei glorreiche Halunken	1966	8.8
10	Der Herr der Ringe: Die Gefährten	2001	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> "Die Verurteilten", "Der Pate", "Der Pate 2", "…
## $ year   <dbl> 1994, 1972, 1974, 2008, 1957, 1993, 2003, 1994,…
## $ rating <dbl> 9.2, 9.1, 9.0, 9.0, 8.9, 8.9, 8.9, 8.8, 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 x 4
##     rank title                                          year rating
##    <int> <chr>                                         <dbl>  <dbl>
##  1     1 Die Verurteilten                               1994    9.2
##  2     2 Der Pate                                       1972    9.1
##  3     3 Der Pate 2                                     1974    9
##  4     4 The Dark Knight                                2008    9
##  5     5 Die zwölf Geschworenen                         1957    8.9
##  6     6 Schindlers Liste                               1993    8.9
##  7     7 Der Herr der Ringe: Die Rückkehr des Königs    2003    8.9
##  8     8 Pulp Fiction                                   1994    8.8
##  9     9 Zwei glorreiche Halunken                       1966    8.8
## 10    10 Der Herr der Ringe: Die Gefährten              2001    8.8
## 11    11 Fight Club                                     1999    8.8
## 12    12 Forrest Gump                                   1994    8.8
## 13    13 Inception                                      2010    8.7
## 14    14 Der Herr der Ringe: Die zwei Türme             2002    8.7
## 15    15 Das Imperium schlägt zurück                    1980    8.7
## 16    16 Matrix                                         1999    8.6
## 17    17 GoodFellas - Drei Jahrzehnte in der Mafia      1990    8.6
## 18    18 Einer flog über das Kuckucksnest               1975    8.6
## 19    19 Die sieben Samurai                             1954    8.6
## 20    20 Sieben                                         1995    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: 84 x 2
##    year     n
##   <dbl> <int>
## 1  1995     8
## 2  2019     7
## 3  1957     6
## 4  2000     6
## 5  2004     6
## 6  2009     6
## # … with 78 more rows
```

Which 1995 movies made the list?

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

```
## # A tibble: 8 x 4
##    rank title                                  year rating
##   <int> <chr>                                <dbl>  <dbl>
## 1    20 Sieben                                1995    8.6
## 2    32 Die üblichen Verdächtigen             1995    8.5
## 3    78 Braveheart                            1995    8.3
## 4    81 Toy Story                             1995    8.3
## 5   123 Heat                                  1995    8.2
## 6   139 Casino                                1995    8.2
## 7   191 Before Sunrise - Zwischenstopp in Wien 1995    8.1
## 8   227 Hass                                  1995    8
```

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

Plot Code