Advanced Web Scraping

Spring 2023

May 04 2023

Get links

```
bow(url = "https://www.imdb.com/search/title/?grou
```

```
<polite session> https://www.imdb.com/search/title
    User-agent: polite R package
    robots.txt: 34 rules are defined for 2 bots
   Crawl delay: 5 sec
  The path is scrapable for this user-agent
```

Get links

```
bow(url = "https://www.imdb.com/search/title/?grou
  scrape()
```

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

Get links

```
bow(url = "https://www.imdb.com/search/title/?grou
  scrape() %>%
  html_elements(css = ".lister-item-header a")
```

```
{xml_nodeset (96)}
 [1] <a href="/title/tt6710474/?ref_=adv_li_tt">Ev
 [2] <a href="/title/tt10366460/?ref_=adv_li_tt">(
 [3] <a href="/title/tt9770150/?ref_=adv_li_tt">Nc
 [4] <a href="/title/tt6751668/?ref_=adv_li_tt">Pa
 [5] <a href="/title/tt6966692/?ref_=adv_li_tt">Gr
 [6] <a href="/title/tt5580390/?ref_=adv_li_tt">Th
 [7] <a href="/title/tt4975722/?ref_=adv_li_tt">Mc
 [8] <a href="/title/tt1895587/?ref_=adv_li_tt">Sp
 [9] <a href="/title/tt2562232/?ref_=adv_li_tt">Bi
[10] <a href="/title/tt2024544/?ref_=adv_li_tt">12
[11] <a href="/title/tt1024648/?ref_=adv_li_tt">Ar
[12] <a href="/title/tt1655442/?ref_=adv_li_tt">Th
[13] <a href="/title/tt1504320/?ref_=adv_li_tt">Th
[14] <a href="/title/tt1010048/?ref_=adv_li_tt">Sl
[15] <a href="/title/tt0887912/?ref_=adv_li_tt">Th
[16] <a href="/title/tt0477348/?ref_=adv_li_tt">Nc
[17] <a href="/title/tt0407887/?ref_=adv_li_tt">Th
[18] <a href="/title/tt0375679/?ref_=adv_li_tt">Cr
[19] <a href="/title/tt0405159/?ref_=adv_li_tt">Mi
[20] <a href="/title/tt0167260/?ref_=adv_li_tt">Th
...
```

Get links

```
bow(url = "https://www.imdb.com/search/title/?gro
  scrape() %>%
  html_elements(css = ".lister-item-header a") %>
  html_attr(name = "href")
```

```
  [1] "/title/tt6710474/?ref_=adv_li_tt"  "/title/
  [3] "/title/tt9770150/?ref_=adv_li_tt"  "/title/
  [5] "/title/tt6966692/?ref_=adv_li_tt"  "/title/
  [7] "/title/tt4975722/?ref_=adv_li_tt"  "/title/
  [9] "/title/tt2562232/?ref_=adv_li_tt"  "/title/
 [11] "/title/tt1024648/?ref_=adv_li_tt"  "/title/
 [13] "/title/tt1504320/?ref_=adv_li_tt"  "/title/
 [15] "/title/tt0887912/?ref_=adv_li_tt"  "/title/
 [17] "/title/tt0407887/?ref_=adv_li_tt"  "/title/
 [19] "/title/tt0405159/?ref_=adv_li_tt"  "/title/
 [21] "/title/tt0299658/?ref_=adv_li_tt"  "/title/
 [23] "/title/tt0172495/?ref_=adv_li_tt"  "/title/
 [25] "/title/tt0138097/?ref_=adv_li_tt"  "/title/
 [27] "/title/tt0116209/?ref_=adv_li_tt"  "/title/
 [29] "/title/tt0109830/?ref_=adv_li_tt"  "/title/
 [31] "/title/tt0105695/?ref_=adv_li_tt"  "/title/
 [33] "/title/tt0099348/?ref_=adv_li_tt"  "/title/
 [35] "/title/tt0095953/?ref_=adv_li_tt"  "/title/
 [37] "/title/tt0091763/?ref_=adv_li_tt"  "/title/
 [39] "/title/tt0086879/?ref_=adv_li_tt"  "/title/
 [41] "/title/tt0083987/?ref_=adv_li_tt"  "/title/
 [43] "/title/tt0081283/?ref_=adv_li_tt"  "/title/
 [45] "/title/tt0077416/?ref_=adv_li_tt"  "/title/
 [47] "/title/tt0075148/?ref_=adv_li_tt"  "/title/
 [49] "/title/tt0071562/?ref_=adv_li_tt"  "/title/
 [51] "/title/tt0068646/?ref_=adv_li_tt"  "/title/
 [53] "/title/tt0066206/?ref_=adv_li_tt"  "/title/
 [55] "/title/tt0063385/?ref_=adv_li_tt"  "/title/
```

Get links

```
bow(url = "https://www.imdb.com/search/title/?gro
  scrape() %>%
  html_elements(css = ".lister-item-header a") %>
  html_attr(name = "href") %>%
  url_absolute(base = "https://www.imdb.com")
```

```
 [1] "https://www.imdb.com/title/tt6710474/?ref_=
 [2] "https://www.imdb.com/title/tt10366460/?ref_
 [3] "https://www.imdb.com/title/tt9770150/?ref_=
 [4] "https://www.imdb.com/title/tt6751668/?ref_=
 [5] "https://www.imdb.com/title/tt6966692/?ref_=
 [6] "https://www.imdb.com/title/tt5580390/?ref_=
 [7] "https://www.imdb.com/title/tt4975722/?ref_=
 [8] "https://www.imdb.com/title/tt1895587/?ref_=
 [9] "https://www.imdb.com/title/tt2562232/?ref_=
[10] "https://www.imdb.com/title/tt2024544/?ref_=
[11] "https://www.imdb.com/title/tt1024648/?ref_=
[12] "https://www.imdb.com/title/tt1655442/?ref_=
[13] "https://www.imdb.com/title/tt1504320/?ref_=
[14] "https://www.imdb.com/title/tt1010048/?ref_=
[15] "https://www.imdb.com/title/tt0887912/?ref_=
[16] "https://www.imdb.com/title/tt0477348/?ref_=
[17] "https://www.imdb.com/title/tt0407887/?ref_=
[18] "https://www.imdb.com/title/tt0375679/?ref_=
[19] "https://www.imdb.com/title/tt0405159/?ref_=
[20] "https://www.imdb.com/title/tt0167260/?ref_=
[21] "https://www.imdb.com/title/tt0299658/?ref_=
[22] "https://www.imdb.com/title/tt0268978/?ref_=
[23] "https://www.imdb.com/title/tt0172495/?ref_=
[24] "https://www.imdb.com/title/tt0169547/?ref_=
[25] "https://www.imdb.com/title/tt0138097/?ref_=
[26] "https://www.imdb.com/title/tt0120338/?ref_=
[27] "https://www.imdb.com/title/tt0116209/?ref_=
[28] "https://www.imdb.com/title/tt0112573/?ref_=
```

Scrape table

```
table_usafacts <- bow(url = "https://usafacts.org/visualizations/covid-vaccine-tracker-states/state/min
  scrape() %>% html_elements(css = "table") %>% html_table() %>% pluck(1)
knitr::kable(table_usafacts, format = "html")
```

State	% of population with at least one dose	% fully vaccinated	% with booster or additional dose
Alabama	64.3%	52.5%	20.1%
Alaska	72%	64.4%	30.8%
Arizona	76.4%	63.8%	29.4%
Arkansas	68.8%	56.1%	24%
California	85.2%	74.2%	41.5%

Click here to take a look at the webpage

Scraping multiple tables

```r
all_url <- "https://finance.yahoo.com/screener/predefined/day_gainers?count=25&offset="
idx <- seq(0, 1050, by = 25)


my_list <- list()

for (i in seq_along(idx)) {
  new_webpage <- read_html(str_glue({all_url}, {idx[i]})) # same as bow(url) %>% scrape()
  table_new <- html_table(new_webpage)[[1]] %>%  # first element of the list
    as_tibble(.name_repair = "unique")  # repairs same column names
  my_list[[i]] <- table_new
}

my_df <- do.call(rbind, my_list)
```

Click here to take a look at the webpage

Multiple tables combined

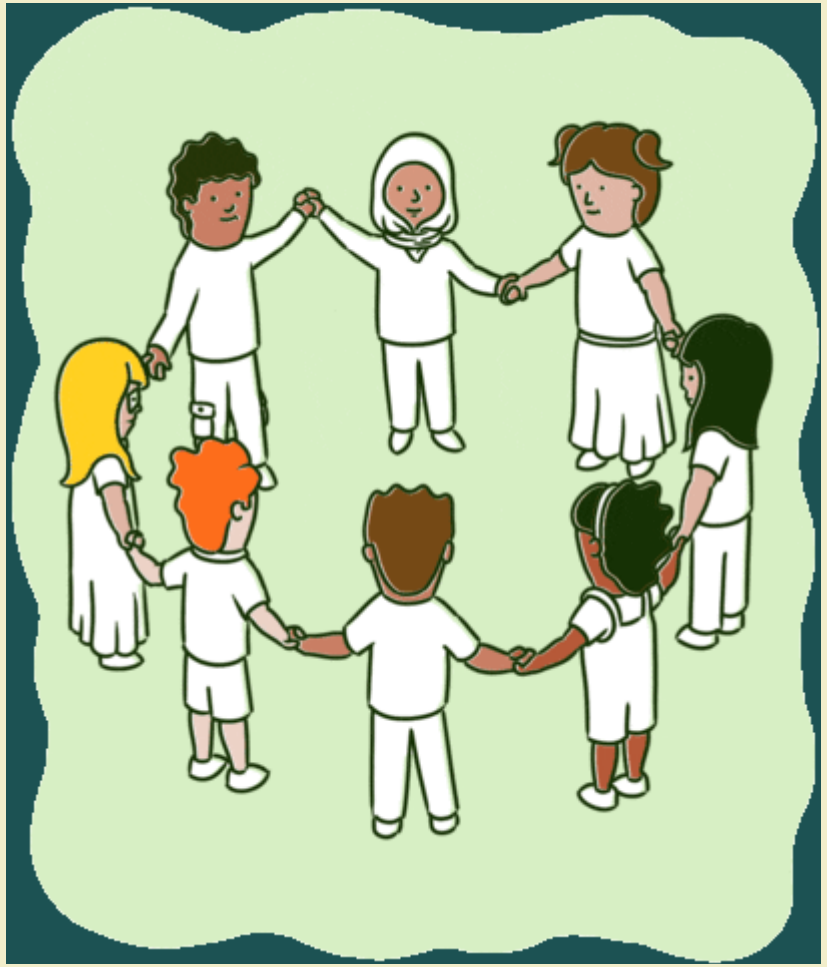
Show 5 entries Search:

	Symbol	Name	Price (Intraday)	Change	% Change	Volume	Avg Vol (3 month)	Market Cap	PE Ratio (TTM)
1	ARNC	Arconic Corporation	28.93	6.38	+28.29%	34.852M	1.859M	2.876B	N/A
2	SHOP	Shopify Inc.	57.3	11.03	+23.84%	88.638M	17.249M	73.955B	N/A
3	GRBK	Green Brick Partners, Inc.	45.76	8.59	+23.11%	1.385M	321,340	2.085B	7.56
4	ITRI	Itron, Inc.	65.6	11.58	+21.44%	794,820	267,383	2.996B	N/A
5	PHJMF	PT Hanjaya Mandala Sampoerna Tbk	0.08	0.012	+17.65%	178,630	91,359	9.46B	N/A

Showing 1 to 5 of 137 entries Previous 1 2 3 4 5 … 28 Next

GROUP ACTIVITY 1

05:00



- *Let's go over to maize server/ local Rstudio and our class moodle*
- *Get the class activity 17.Rmd file*
- *Work on activity 1*

Tidy further

```
df_movies
```

```
# A tibble: 6,389 × 6
   ...1  ReleaseDate  Movie
   <chr> <chr>        <chr>
 1 1     Dec 9, 2022  Avatar: The Way o
 2 2     Apr 23, 2019 Avengers: Endgame
 3 3     May 20, 2011 Pirates of the Ca
 4 4     Apr 22, 2015 Avengers: Age of
 5 5     May 17, 2023 Fast X
 6 6     Dec 16, 2015 Star Wars Ep. VII
 7 7     Apr 25, 2018 Avengers: Infinit
 8 8     May 24, 2007 Pirates of the Ca
 9 9     Nov 13, 2017 Justice League
10 10    Oct 6, 2015  Spectre
# … with 6,379 more rows, and abbreviat
#   ²DomesticGross, ³WorldwideGross
```

Tidy further

```
df_movies %>%
  rename(ID = `...1`)
```

```
# A tibble: 6,389 × 6
   ID    ReleaseDate  Movie
   <chr> <chr>        <chr>
 1 1     Dec 9, 2022  Avatar: The Way o
 2 2     Apr 23, 2019 Avengers: Endgame
 3 3     May 20, 2011 Pirates of the Ca
 4 4     Apr 22, 2015 Avengers: Age of
 5 5     May 17, 2023 Fast X
 6 6     Dec 16, 2015 Star Wars Ep. VII
 7 7     Apr 25, 2018 Avengers: Infinit
 8 8     May 24, 2007 Pirates of the Ca
 9 9     Nov 13, 2017 Justice League
10 10    Oct 6, 2015  Spectre
# … with 6,379 more rows, and abbreviat
#   ²DomesticGross, ³WorldwideGross
```

Tidy further

```
df_movies %>%
  rename(ID = `...1`) %>%
  mutate(ProductionBudget = parse_number(ProductionBudget),
         DomesticGross = parse_number(DomesticGross),
         WorldwideGross = parse_number(WorldwideGross),
         ReleaseDate = mdy(ReleaseDate),
         ReleaseDate = replace_na(ReleaseDate, make_date()),
         MonthOfRelease = month(ReleaseDate, label = TRUE),
         YearOfRelease = year(ReleaseDate))
```

```
# A tibble: 6,389 × 8
   ID    ReleaseDate Movie
   <chr> <date>      <chr>
 1 1     2022-12-09  Avatar: The Way of.
 2 2     2019-04-23  Avengers: Endgame
 3 3     2011-05-20  Pirates of the Car.
 4 4     2015-04-22  Avengers: Age of U.
 5 5     2023-05-17  Fast X
 6 6     2015-12-16  Star Wars Ep. VII:.
 7 7     2018-04-25  Avengers: Infinity.
 8 8     2007-05-24  Pirates of the Car.
 9 9     2017-11-13  Justice League
10 10    2015-10-06  Spectre
# … with 6,379 more rows, and abbreviat
#   ²DomesticGross, ³WorldwideGross, ⁴M
```

Tidy further

```
df_movies %>%
  rename(ID = `...1`) %>%
  mutate(ProductionBudget = parse_number(ProductionBudget),
         DomesticGross = parse_number(DomesticGross),
         WorldwideGross = parse_number(WorldwideGross),
         ReleaseDate = mdy(ReleaseDate),
         ReleaseDate = replace_na(ReleaseDate, make_date()),
         MonthOfRelease = month(ReleaseDate, label = TRUE),
         YearOfRelease = year(ReleaseDate)) %>%
  select(MonthOfRelease, DomesticGross)
```

```
# A tibble: 6,389 × 2
   MonthOfRelease DomesticGross
   <ord>                  <dbl>
 1 Dec                683978730
 2 Apr                858373000
 3 May                241071802
 4 Apr                459005868
 5 May                        0
 6 Dec                936662225
 7 Apr                678815482
 8 May                309420425
 9 Nov                229024295
10 Oct                200074175
# … with 6,379 more rows
```

Tidy further

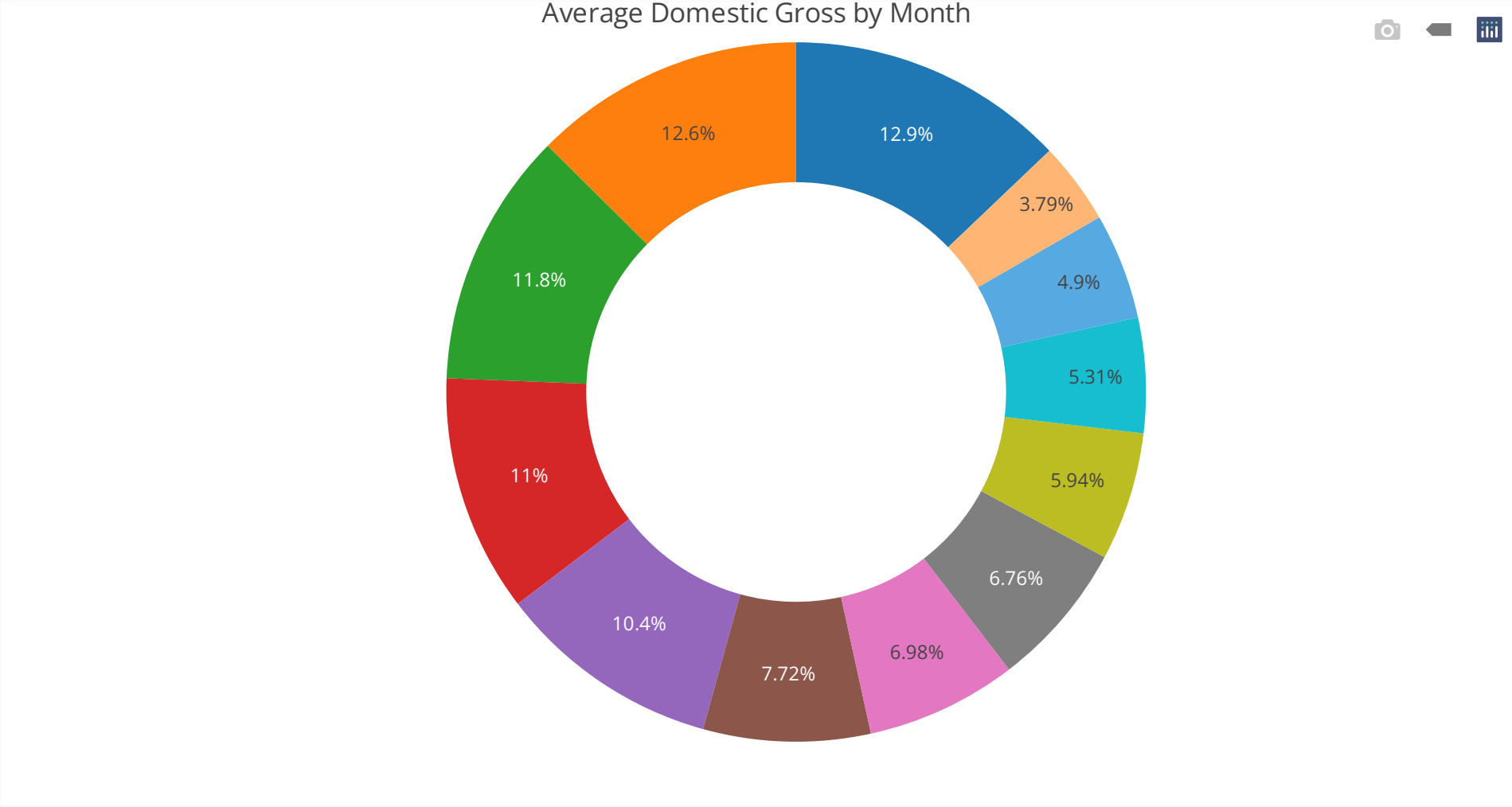
```
df_movies %>%
  rename(ID = `...1`) %>%
  mutate(ProductionBudget = parse_number(ProductionBudget),
         DomesticGross = parse_number(DomesticGross),
         WorldwideGross = parse_number(WorldwideGross),
         ReleaseDate = mdy(ReleaseDate),
         ReleaseDate = replace_na(ReleaseDate, make_date()),
         MonthOfRelease = month(ReleaseDate, label = TRUE),
         YearOfRelease = year(ReleaseDate)) %>%
  select(MonthOfRelease, DomesticGross) %>%
  group_by(MonthOfRelease)
```

```
# A tibble: 6,389 × 2
# Groups:   MonthOfRelease [12]
   MonthOfRelease DomesticGross
   <ord>                  <dbl>
 1 Dec                683978730
 2 Apr                858373000
 3 May                241071802
 4 Apr                459005868
 5 May                        0
 6 Dec                936662225
 7 Apr                678815482
 8 May                309420425
 9 Nov                229024295
10 Oct                200074175
# … with 6,379 more rows
```

Tidy further

```
df_movies %>%
  rename(ID = `...1`) %>%
  mutate(ProductionBudget = parse_number(ProductionBudget),
         DomesticGross = parse_number(DomesticGross),
         WorldwideGross = parse_number(WorldwideGross),
         ReleaseDate = mdy(ReleaseDate),
         ReleaseDate = replace_na(ReleaseDate, make_date()),
         MonthOfRelease = month(ReleaseDate, label = TRUE),
         YearOfRelease = year(ReleaseDate)) %>%
  select(MonthOfRelease, DomesticGross) %>%
  group_by(MonthOfRelease) %>%
  summarize(AverageByMonth = mean(DomesticGross))
```

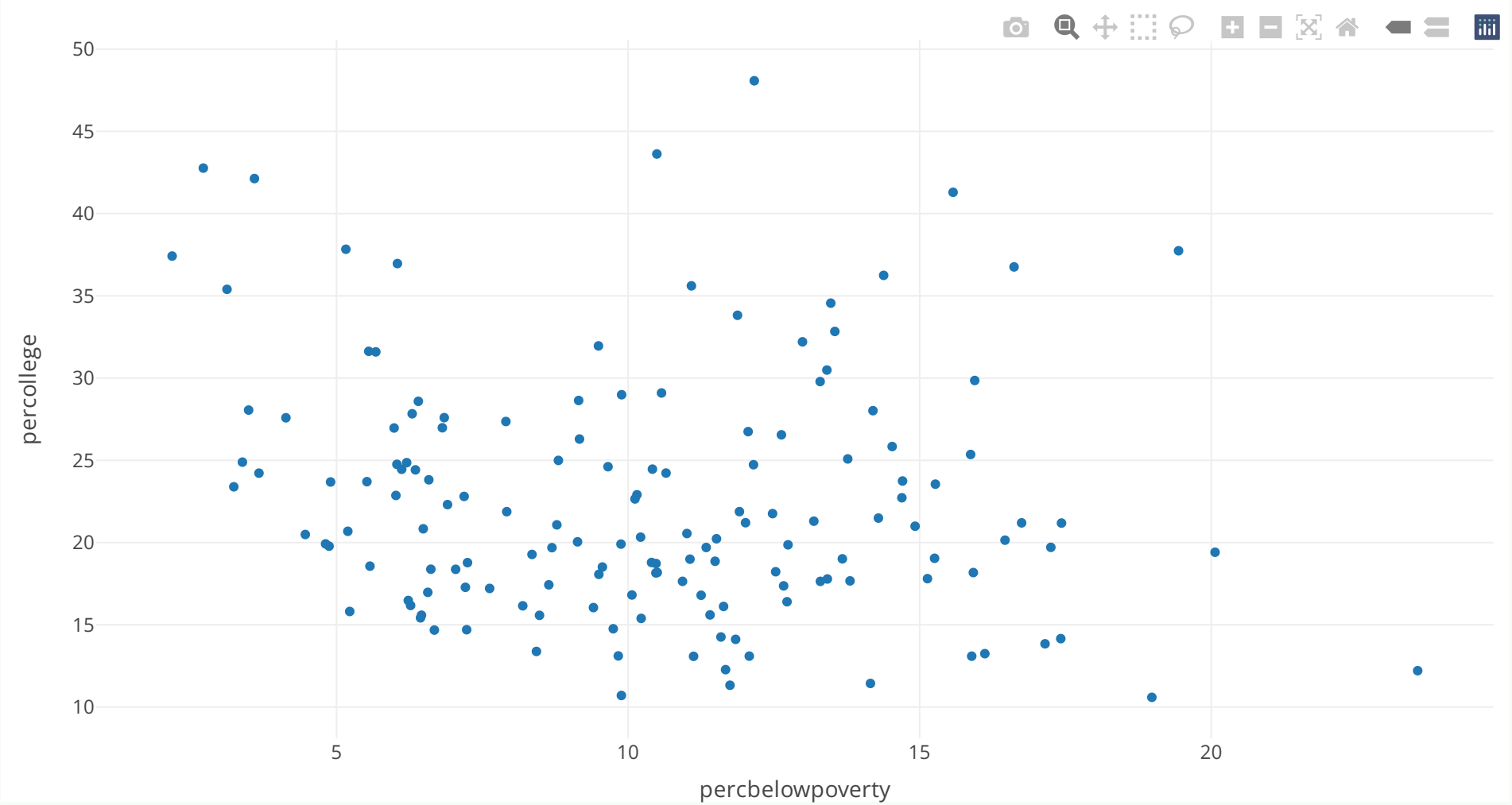
```
# A tibble: 12 × 2
   MonthOfRelease AverageByMonth
   <ord>                   <dbl>
 1 Jan                  19051874.
 2 Feb                  35061089.
 3 Mar                  38753002.
 4 Apr                  33950505.
 5 May                  63134312.
 6 Jun                  64686623.
 7 Jul                  55150729.
 8 Aug                  29806928.
 9 Sep                  24612120.
10 Oct                  26677281.
11 Nov                  51989631.
12 Dec                  59260477.
```


Average Domestic Gross by Month
12.9%
3.79%
4.9%
5.31%
5.94%
6.76%
6.98%
7.72%
10.4%
11%
11.8%
12.6%

Interactive visualizations using ggplotly

```
midwest %>% as_tibble()
# A tibble: 437 × 28
     PID county    state  area poptotal popden…¹ popwh…² popbl…³ popam…⁴ popas…⁵
   <int> <chr>     <chr> <dbl>    <int>    <dbl>   <int>   <int>   <int>   <int>
 1   561 ADAMS     IL    0.052    66090    1271.   63917    1702      98     249
 2   562 ALEXANDER IL    0.014    10626     759     7054    3496      19      48
 3   563 BOND      IL    0.022    14991     681.   14477     429      35      16
 4   564 BOONE     IL    0.017    30806    1812.   29344     127      46     150
 5   565 BROWN     IL    0.018     5836     324.    5264     547      14       5
 6   566 BUREAU    IL    0.05     35688     714.   35157      50      65     195
 7   567 CALHOUN   IL    0.017     5322     313.    5298       1       8      15
 8   568 CARROLL   IL    0.027    16805     622.   16519     111      30      61
 9   569 CASS      IL    0.024    13437     560.   13384      16       8      23
10   570 CHAMPAIGN IL    0.058   173025    2983.  146506   16559     331    8033
# … with 427 more rows, 18 more variables: popother <int>, percwhite <dbl>,
```

Interactive visualizations using `ggplotly`



Interactive visualizations using `ggplotly`

```
mtcars %>% as_tibble() %>% head()
# A tibble: 6 × 11
    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21       6   160   110  3.9   2.62  16.5     0     1     4     4
2  21       6   160   110  3.9   2.88  17.0     0     1     4     4
3  22.8     4   108    93  3.85  2.32  18.6     1     1     4     1
4  21.4     6   258   110  3.08  3.22  19.4     1     0     3     1
5  18.7     8   360   175  3.15  3.44  17.0     0     0     3     2
6  18.1     6   225   105  2.76  3.46  20.2     1     0     3     1
```

Interactive visualizations using ggplotly

```
mtcars %>% as_tibble() %>% head()
# A tibble: 6 × 11
    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21       6   160   110  3.9   2.62  16.5     0     1     4     4
2  21       6   160   110  3.9   2.88  17.0     0     1     4     4
3  22.8     4   108    93  3.85  2.32  18.6     1     1     4     1
4  21.4     6   258   110  3.08  3.22  19.4     1     0     3     1
5  18.7     8   360   175  3.15  3.44  17.0     0     0     3     2
6  18.1     6   225   105  2.76  3.46  20.2     1     0     3     1
```

```
gp = mtcars %>%
  mutate(amFactor = factor(am, labels = c('auto', 'manual')),
         hovertext = paste(wt, mpg, amFactor)) %>%
  arrange(wt) %>%
  ggplot(aes(x = wt, y = mpg, color = amFactor)) +
  geom_smooth(se = F) +
  geom_point(aes(color = amFactor)) + theme_economist_white()
```

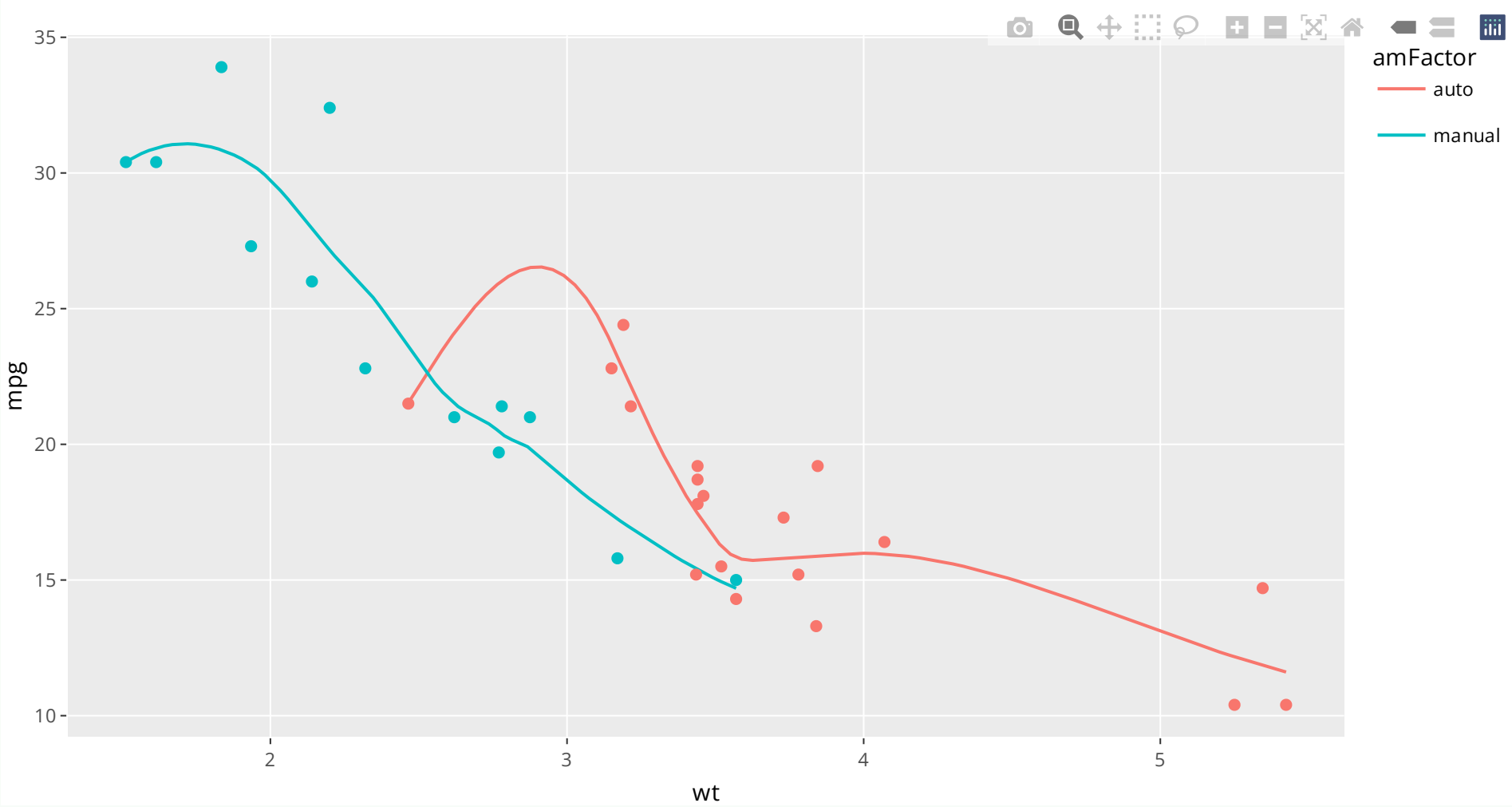
Interactive visualizations using ggplotly

```
mtcars %>% as_tibble() %>% head()
# A tibble: 6 × 11
    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21       6   160   110  3.9   2.62  16.5     0     1     4     4
2  21       6   160   110  3.9   2.88  17.0     0     1     4     4
3  22.8     4   108    93  3.85  2.32  18.6     1     1     4     1
4  21.4     6   258   110  3.08  3.22  19.4     1     0     3     1
5  18.7     8   360   175  3.15  3.44  17.0     0     0     3     2
6  18.1     6   225   105  2.76  3.46  20.2     1     0     3     1
```

```
gp = mtcars %>%
  mutate(amFactor = factor(am, labels = c('auto', 'manual')),
         hovertext = paste(wt, mpg, amFactor)) %>%
  arrange(wt) %>%
  ggplot(aes(x = wt, y = mpg, color = amFactor)) +
  geom_smooth(se = F) +
  geom_point(aes(color = amFactor)) + theme_economist_white()
```

```
ggplotly()
```

Interactive visualizations using `ggplotly`



DT: Interactive Data Tables

```
library(ggplot2movies)
movies %>%
  select(1:6) %>%
  filter(rating > 8, !is.na(budget), votes > 1000) %>%
  datatable(fillContainer = FALSE, options = list(pageLength = 6))
```

DT: Interactive Data Tables

```
library(ggplot2movies)
movies %>%
  select(1:6) %>%
  filter(rating > 8, !is.na(budget), votes > 1000) %>%
  datatable(fillContainer = FALSE, options = list(pageLength = 6))
```

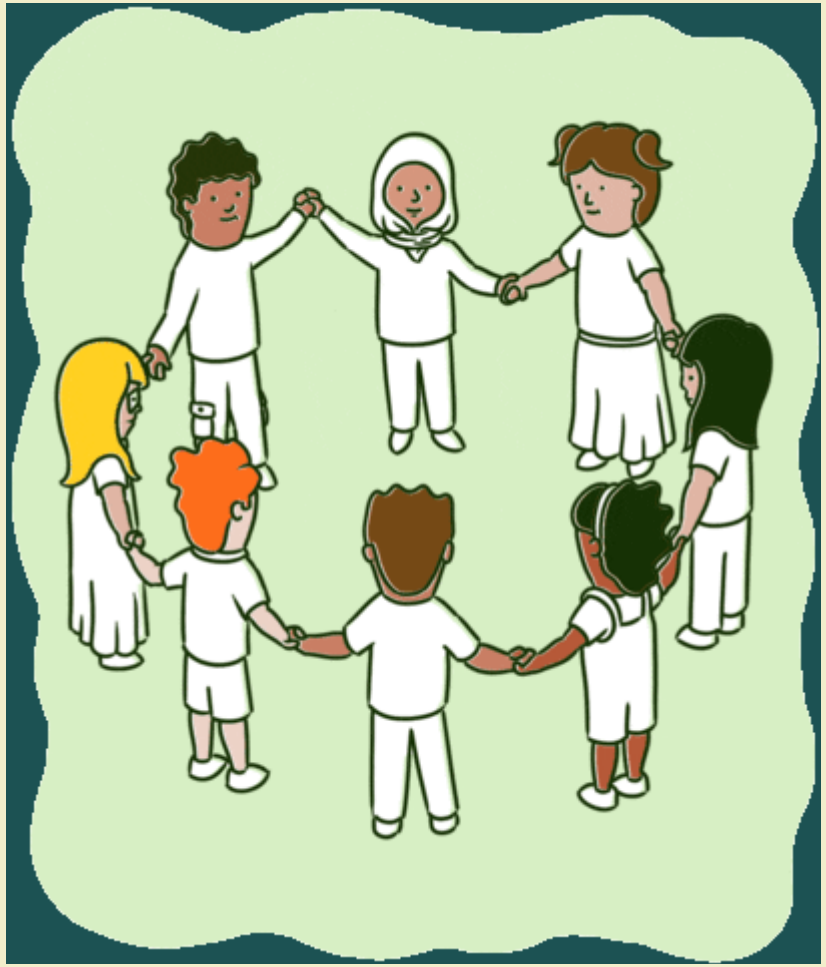
Show 6 entries Search:

	title	year	length	budget	rating	votes
1	12 Angry Men	1957	96	340000	8.7	29278
2	2001: A Space Odyssey	1968	156	10500000	8.3	64982
3	Adventures of Robin Hood, The	1938	102	1900000	8.2	7359
4	Alien	1979	116	11000000	8.3	63400
5	Aliens	1986	154	18500000	8.3	63961
6	All Quiet on the Western Front	1930	147	1200000	8.2	6835

Showing 1 to 6 of 149 entries Previous 1 2 3 4 5 … 25 Next

GROUP ACTIVITY 2

15:00



- Work on activity 2
- Ask me questions