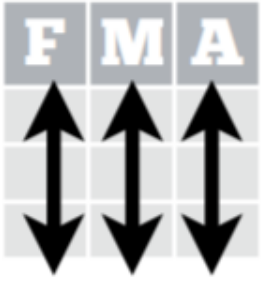

tidyverse
www.rstudio.com

# Tidy data

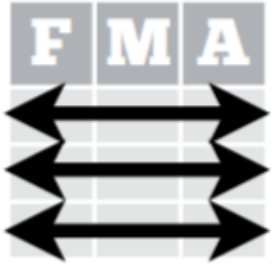

# Tidy data


In a tidy data set:




Each **variable** is saved in its own **column**

&

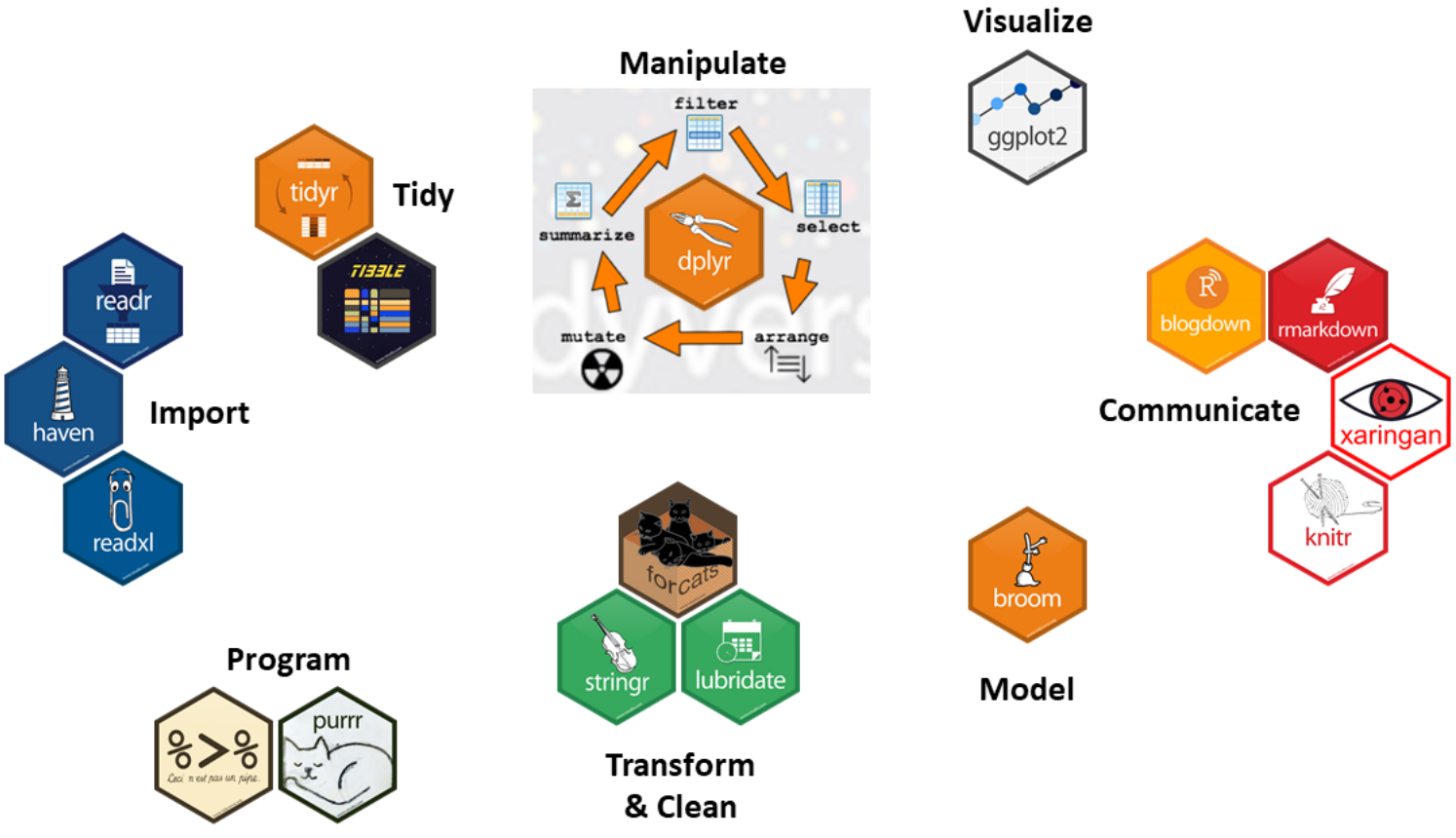



Each **observation** is saved in its own **row**

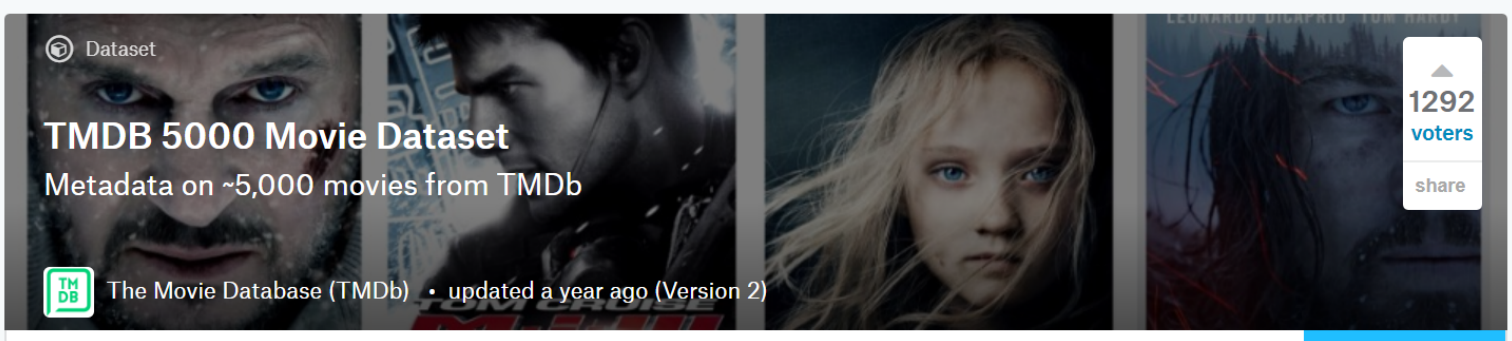
# www.tidyverse.org


```
#install.packages("tidyverse")

library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.0.0.9000     ✔ purrr   0.2.5
## ✔ tibble  1.4.2          ✔ dplyr   0.7.6
## ✔ tidyr   0.8.1          ✔ stringr 1.3.1
## ✔ readr   1.1.1          ✔ forcats 0.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```


Visualize
ggplot2
Manipulate
filter
summarize
select
dplyr
mutate
arrange
tidyr
Tidy
tibble
readr
haven
readxl
Import
blogdown
rmarkdown
xaringan
knitr
Communicate
broom
Model
forcats
stringr
lubridate
Transform
& Clean
purrr
Program

# Data


Dataset
TMDB 5000 Movie Dataset
Metadata on ~5,000 movies from TMDb
The Movie Database (TMDb) • updated a year ago (Version 2)
1292
voters
share
Data
Overview
Kernels (1,417)
Discussion (50)
Activity
Download (9 MB)
New Kernel
Data (9 MB)
API
kaggle datasets download -d tmdb/tmdb-movie-meta…
?
Download All
Data Sources
tmdb_5000_credits.c... 4803 x 4
tmdb_5000_movies... 4803 x 20
About this file
Edit
Help us describe this file
Columns
Edit
movie_id
title
cast Mark Wahlberg
crew

# Import data


```
data_movies <- read_csv("data/tmdb_5000_movies.csv")
data_credits <- read_csv("data/tmdb_5000_credits.csv")

data_movies
```

# Tibble


```
data_movies <- read_csv("data/tmdb_5000_movies.csv")
data_credits <- read_csv("data/tmdb_5000_credits.csv")

data_movies
```

```
## # A tibble: 4,803 x 20
##     budget genres homepage     id keywords original_langua… original_title
##      <int> <chr>  <chr>     <int> <chr>    <chr>            <chr>
##  1  2.37e8 "[{\"… http://…  19995 "[{\"id… en               Avatar
##  2  3.00e8 "[{\"… http://…    285 "[{\"id… en               Pirates of th…
##  3  2.45e8 "[{\"… http://… 206647 "[{\"id… en               Spectre
##  4  2.50e8 "[{\"… http://…  49026 "[{\"id… en               The Dark Knig…
##  5  2.60e8 "[{\"… http://…  49529 "[{\"id… en               John Carter
##  6  2.58e8 "[{\"… http://…    559 "[{\"id… en               Spider-Man 3
##  7  2.60e8 "[{\"… http://…  38757 "[{\"id… en               Tangled
##  8  2.80e8 "[{\"… http://…  99861 "[{\"id… en               Avengers: Age…
##  9  2.50e8 "[{\"… http://…    767 "[{\"id… en               Harry Potter …
## 10  2.50e8 "[{\"… http://… 209112 "[{\"id… en               Batman v Supe…
## # ... with 4,793 more rows, and 13 more variables: overview <chr>,
## #   popularity <dbl>, production_companies <chr>,
## #   production_countries <chr>, release_date <date>, revenue <dbl>,
## #   runtime <int>, spoken_languages <chr>, status <chr>, tagline <chr>,
## #   title <chr>, vote_average <dbl>, vote_count <int>
```

# Pipe data

# Pipe data

- old way : **verb(subject, complements)**

```
head(data_movies,3)
```

# Pipe data

- old way : **verb(subject, complements)**

```
head(data_movies,3)
```


- pipe way : **subject %>% verb(complements)**

```
data_movies %>% head(3)
```

```
## # A tibble: 3 x 20
##    budget genres homepage     id keywords  original_
##     <int> <chr>  <chr>     <int> <chr>     <chr>
## 1  2.37e8 "[{\"… http://…  19995 "[{\"id\… en
## 2  3.00e8 "[{\"… http://…    285 "[{\"id\… en
## 3  2.45e8 "[{\"… http://… 206647 "[{\"id\… en
## # ... with 13 more variables: overview <chr>, popula
## #   production_companies <chr>, production_countries
## #   release_date <date>, revenue <dbl>, runtime <int
## #   spoken_languages <chr>, status <chr>, tagline <c
## #   vote_average <dbl>, vote_count <int>
```

# Functional programming


```
both <- list("data_movies" = data_movies, "data_credits" = data_credits)

both %>%
  map(names)
```

```
## $data_movies
##  [1] "budget"               "genres"               "homepage"
##  [4] "id"                   "keywords"             "original_language"
##  [7] "original_title"       "overview"             "popularity"
## [10] "production_companies" "production_countries" "release_date"
## [13] "revenue"              "runtime"              "spoken_languages"
## [16] "status"               "tagline"              "title"
## [19] "vote_average"         "vote_count"
##
## $data_credits
## [1] "movie_id" "title"    "cast"     "crew"
```

# Functional programming


```
both <- list("data_movies" = data_movies, "data_credits" = data_credits)

both %>%
  map(names)
```

```
## $data_movies
##  [1] "budget"               "genres"               "homepage"
##  [4] "id"                   "keywords"             "original_language"
##  [7] "original_title"       "overview"             "popularity"
## [10] "production_companies" "production_countries" "release_date"
## [13] "revenue"              "runtime"              "spoken_languages"
## [16] "status"               "tagline"              "title"
## [19] "vote_average"         "vote_count"
##
## $data_credits
## [1] "movie_id" "title"    "cast"     "crew"
```

# Functional programming

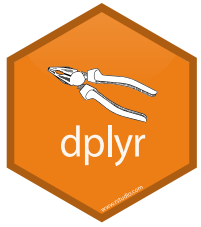

```
both <- list("data_movies" = data_movies, "data_credits" = data_credits)

both %>%
  map(names)
```

```
## $data_movies
##  [1] "budget"                "genres"                "homepage"
##  [4] "id"                    "keywords"              "original_language"
##  [7] "original_title"        "overview"              "popularity"
## [10] "production_companies"  "production_countries"  "release_date"
## [13] "revenue"               "runtime"               "spoken_languages"
## [16] "status"                "tagline"               "title"
## [19] "vote_average"          "vote_count"
##
## $data_credits
## [1] "movie_id" "title"    "cast"     "crew"
```

- => Rename **id** to **movie_id** + join both

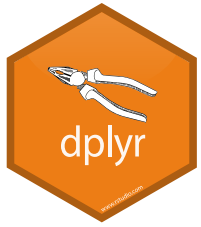

JOIN US!

# Rename variables




```
both <- data_movies %>%
  rename(movie_id = id) %>%
  left_join(data_credits)
```

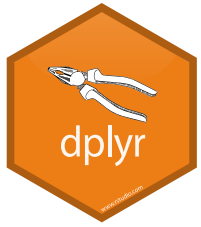
# Join data




```
both <- data_movies %>%
  rename(movie_id = id) %>%
  left_join(data_credits)
```

```
## Joining, by = c("movie_id", "title")
```

# Join data




```
both <- data_movies %>%
  rename(movie_id = id) %>%
  left_join(data_credits)
```

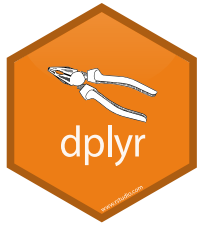
```
## Joining, by = c("movie_id", "title")
```

```
both %>% names()
```

```
##  [1] "budget"              "genres"              "homepage"
##  [4] "movie_id"            "keywords"            "original_language"
##  [7] "original_title"      "overview"            "popularity"
## [10] "production_companies" "production_countries" "release_date"
## [13] "revenue"             "runtime"             "spoken_languages"
## [16] "status"              "tagline"             "title"
## [19] "vote_average"        "vote_count"          "cast"
## [22] "crew"
```

| budget | genres | homepage |
|---|---|---|
| 2.37e+08 | [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}] | http://www.avatarmovie.com/ |

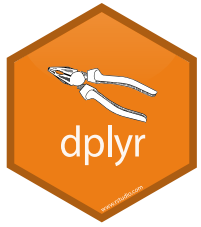
# Select variables




```
data_movies <- data_movies %>%
  select(title, budget, vote_average, production_companies)
```

| title | budget | vote_average | production_companies |
|---|---|---|---|
| Avatar | 2.37e+08 | 7.2 | [{"name": "Ingenious Film Partners", "id": 289}, {"name": "Twentieth Century Fox Film Corporation", "id": 306}, {"name": "Dune Entertainment", "id": 444}, {"name": "Lightstorm Entertainment", "id": 574}] |
| Pirates of the Caribbean: At World's End | 3.00e+08 | 6.9 | [{"name": "Walt Disney Pictures", "id": 2}, {"name": "Jerry Bruckheimer Films", "id": 130}, {"name": "Second Mate Productions", "id": 19936}] |
| Spectre | 2.45e+08 | 6.3 | [{"name": "Columbia Pictures", "id": 5}, {"name": "Danjaq", "id": 10761}, {"name": "B24", "id": 69434}] |

# Add a column

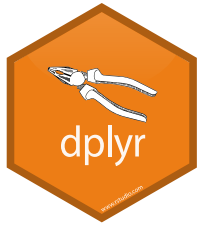



```
data_movies <- data_movies %>%
  mutate(Production = case_when(
    str_detect(production_companies,"Disney") ~ "Disney",
    str_detect(production_companies,"Marvel") ~ "Marvel",
    str_detect(production_companies,"DC") ~ "DC",
    TRUE ~ "Other"
  ))
```

# String manipulation


```
data_movies <- data_movies %>%
  mutate(Production = case_when(
    str_detect(production_companies,"Disney") ~ "Disney",
    str_detect(production_companies,"Marvel") ~ "Marvel",
    str_detect(production_companies,"DC") ~ "DC",
    TRUE ~ "Other"
  ))
```

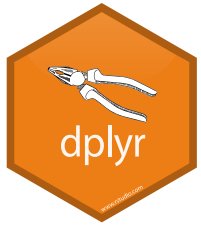
# Remove a column




```
data_movies <- data_movies %>%
  mutate(Production = case_when(
    str_detect(production_companies,"Disney") ~ "Disney",
    str_detect(production_companies,"Marvel") ~ "Marvel",
    str_detect(production_companies,"DC") ~ "DC",
    TRUE ~ "Other"
  )) %>%
  select(-production_companies)
```

| title | budget | vote_average | Production |
|---|---|---|---|
| Avatar | 2.37e+08 | 7.2 | Other |
| Pirates of the Caribbean: At World's End | 3.00e+08 | 6.9 | Disney |
| Spectre | 2.45e+08 | 6.3 | Other |
| The Dark Knight Rises | 2.50e+08 | 7.6 | DC |
| John Carter | 2.60e+08 | 6.1 | Disney |
| Spider-Man 3 | 2.58e+08 | 5.9 | Marvel |
| Tangled | 2.60e+08 | 7.4 | Disney |

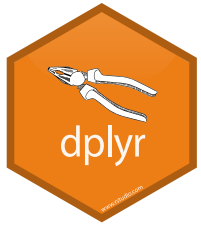
# Filter




Keep Disney, Marvel and DC movies

```
data_movies2 <- data_movies %>%
  filter(Production != "Other")
```

# Filter




Keep Disney, Marvel and DC movies

```
data_movies2 <- data_movies %>%
  filter(Production != "Other")
```

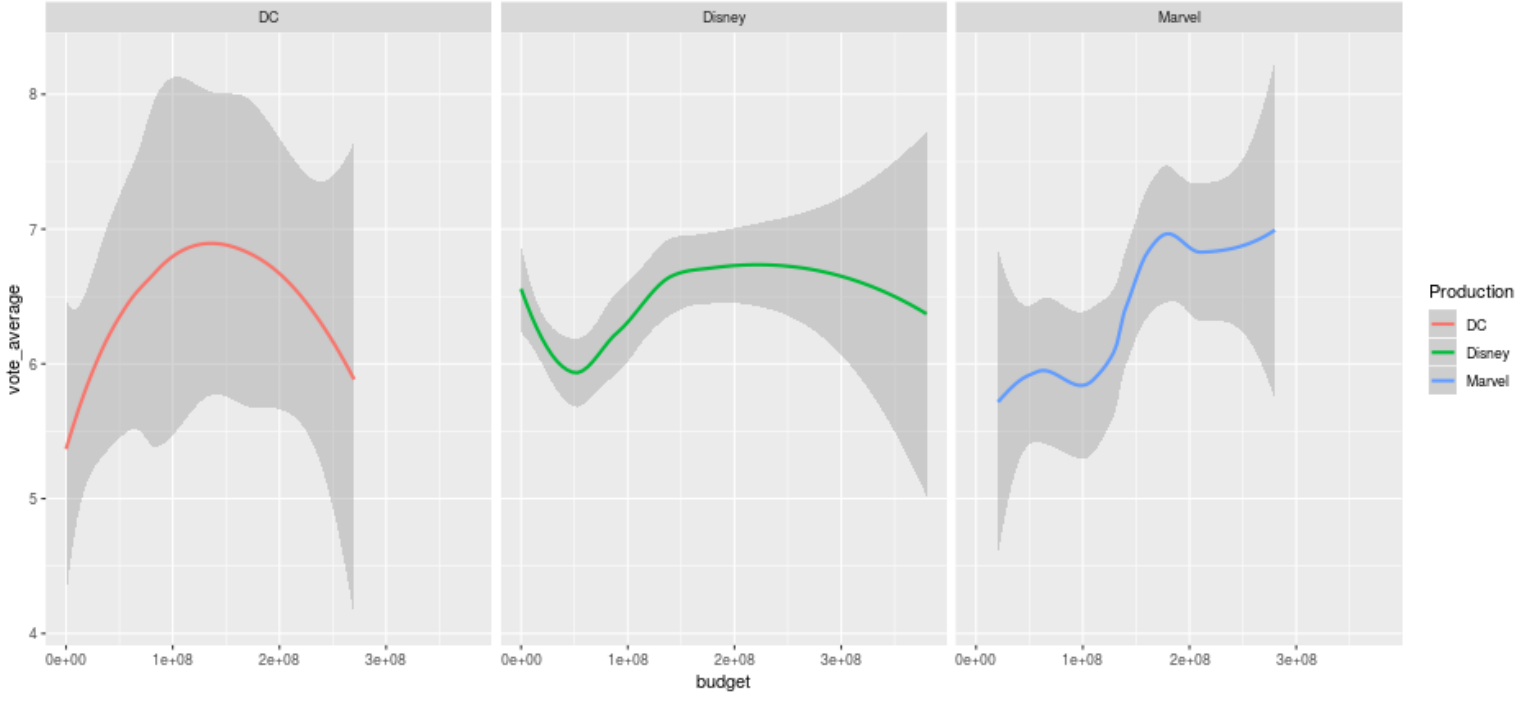
# group_by, summarise, arrange

```
data_movies2 %>%
  group_by(Production) %>%
  summarise(mean_vote = mean(vote_average)) %>%
  arrange(desc(mean_vote))
```

```
## # A tibble: 3 x 2
##   Production mean_vote
##   <chr>          <dbl>
## 1 Marvel          6.41
## 2 Disney          6.35
## 3 DC              6.16
```

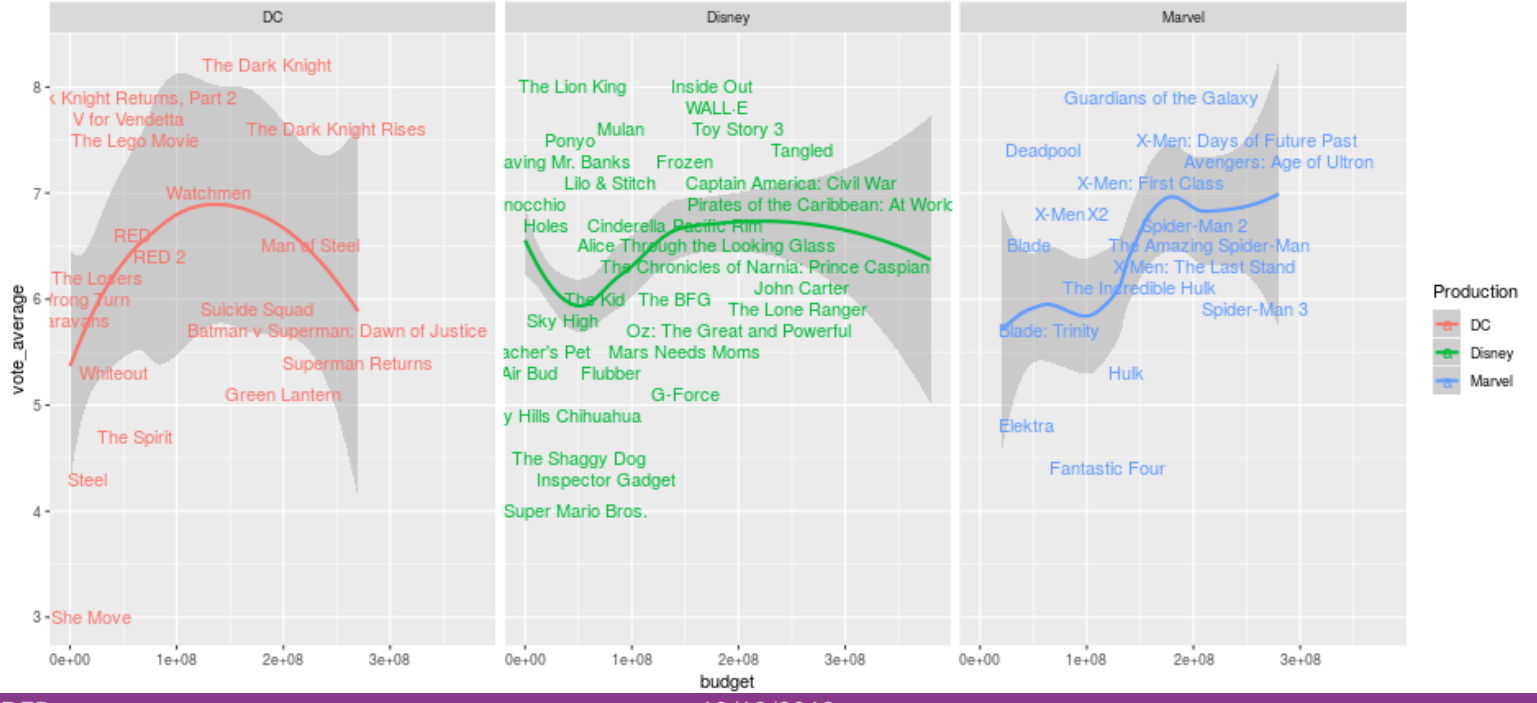
# Visualize data

```
data_movies2 %>%
  ggplot(aes(x = budget, y = vote_average,
             col = Production, label = title)) +
  geom_smooth() +
  facet_wrap(~Production)
```



# Visualize data

```
data_movies2 %>%
  ggplot(aes(x = budget, y = vote_average,
             col = Production, label = title)) +
  geom_smooth() +
  facet_wrap(~Production) +
  geom_text(check_overlap = TRUE)
```



# Thanks!

- Cheat sheets


Slides created via the R package **xaringan** with the **R-Ladies theme**