hw_06.R

aaron

Wed Mar 21 19:46:34 2018

```
library(yrbss)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
library(ggplot2)
library(reshape2)
library(ggplot2movies)
### Your turn

## * Given the dataset `iris`:
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
## * Get a subset containing only `Species` `"versicolor"`,
##   such that `Sepal.Width` is less than $2.5$.

## Begin solution:
iris %>%
  subset(Species == "versicolor" & Sepal.Width < 2.5)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 54          5.5         2.3          4.0         1.3 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
```

```r
## End solution


## * Get a subset containing only `Species` `"versicolor"` and `"virginica"`,
##   such that `Sepal.Width` is between $2.5$ and $3.2$. Keep only columns `Species`
##   and `Sepal.Width` (in that order).

## Begin solution
iris %>%
  subset(Species == "versicolor" | Species == "virginica") %>%
  subset(Sepal.Width >= 2.5 & Sepal.Width <= 3.2) %>%
  select(Species, Sepal.Width)
```

```
##        Species Sepal.Width
## 51  versicolor         3.2
## 52  versicolor         3.2
## 53  versicolor         3.1
## 55  versicolor         2.8
## 56  versicolor         2.8
## 59  versicolor         2.9
## 60  versicolor         2.7
## 62  versicolor         3.0
## 64  versicolor         2.9
## 65  versicolor         2.9
## 66  versicolor         3.1
## 67  versicolor         3.0
## 68  versicolor         2.7
## 70  versicolor         2.5
## 71  versicolor         3.2
## 72  versicolor         2.8
## 73  versicolor         2.5
## 74  versicolor         2.8
## 75  versicolor         2.9
## 76  versicolor         3.0
## 77  versicolor         2.8
## 78  versicolor         3.0
## 79  versicolor         2.9
## 80  versicolor         2.6
## 83  versicolor         2.7
## 84  versicolor         2.7
## 85  versicolor         3.0
## 87  versicolor         3.1
## 89  versicolor         3.0
## 90  versicolor         2.5
## 91  versicolor         2.6
## 92  versicolor         3.0
## 93  versicolor         2.6
## 95  versicolor         2.7
## 96  versicolor         3.0
## 97  versicolor         2.9
## 98  versicolor         2.9
## 99  versicolor         2.5
## 100 versicolor         2.8
## 102  virginica         2.7
## 103  virginica         3.0
## 104  virginica         2.9
## 105  virginica         3.0
## 106  virginica         3.0
## 107  virginica         2.5
## 108  virginica         2.9
## 109  virginica         2.5
## 111  virginica         3.2
## 112  virginica         2.7
## 113  virginica         3.0
## 114  virginica         2.5
## 115  virginica         2.8
## 116  virginica         3.2
## 117  virginica         3.0
## 119  virginica         2.6
## 121  virginica         3.2
## 122  virginica         2.8
## 123  virginica         2.8
## 124  virginica         2.7
## 126  virginica         3.2
## 127  virginica         2.8
## 128  virginica         3.0
## 129  virginica         2.8
## 130  virginica         3.0
## 131  virginica         2.8
## 133  virginica         2.8
## 134  virginica         2.8
## 135  virginica         2.6
## 136  virginica         3.0
## 138  virginica         3.1
## 139  virginica         3.0
## 140  virginica         3.1
## 141  virginica         3.1
## 142  virginica         3.1
## 143  virginica         2.7
## 144  virginica         3.2
## 146  virginica         3.0
## 147  virginica         2.5
## 148  virginica         3.0
## 150  virginica         3.0
```

```r
## End solution


## * Calculate the means for each of the 4 numerical variables.

## Begin solution
iris %>%
  summarise(n = n(),
            Sepal.Length_mean = mean(Sepal.Length, na.rm = TRUE),
            Sepal.Width_mean = mean(Sepal.Width, na.rm = TRUE),
            Petal.Length_mean = mean(Petal.Length, na.rm = TRUE),
            Petal.Width_mean = mean(Petal.Width, na.rm = TRUE))
```

```
##     n Sepal.Length_mean Sepal.Width_mean Petal.Length_mean
## 1 150          5.843333         3.057333             3.758
##   Petal.Width_mean
## 1         1.199333
```

```
## End solution


## * Include the medians to the previous problem.

## Begin solution
iris %>%
  summarise(n = n(),
            Sepal.Length_mean = mean(Sepal.Length, na.rm = TRUE),
            Sepal.Width_mean = mean(Sepal.Width, na.rm = TRUE),
            Petal.Length_mean = mean(Petal.Length, na.rm = TRUE),
            Petal.Width_mean = mean(Petal.Width, na.rm = TRUE),
            Sepal.Length_median = median(Sepal.Length, na.rm = TRUE),
            Sepal.Width_median = median(Sepal.Width, na.rm = TRUE),
            Petal.Length_median = median(Petal.Length, na.rm = TRUE),
            Petal.Width_median = median(Petal.Width, na.rm = TRUE))
```

```
##     n Sepal.Length_mean Sepal.Width_mean Petal.Length_mean
## 1 150          5.843333         3.057333             3.758
##   Petal.Width_mean Sepal.Length_median Sepal.Width_median
## 1         1.199333                 5.8                  3
##   Petal.Length_median Petal.Width_median
## 1                4.35                1.3
```

```
## End solution

## * Calculate the means for each of the 4 numerical variables,
##  by `Species`.

## Begin solution
iris %>%
  group_by(Species) %>%
  summarise(n = n(),
            Sepal.Length_mean = mean(Sepal.Length, na.rm = TRUE),
            Sepal.Width_mean = mean(Sepal.Width, na.rm = TRUE),
            Petal.Length_mean = mean(Petal.Length, na.rm = TRUE),
            Petal.Width_mean = mean(Petal.Width, na.rm = TRUE))
```

```
## # A tibble: 3 x 6
##      Species     n Sepal.Length_mean Sepal.Width_mean Petal.Length_mean
##       <fctr> <int>             <dbl>            <dbl>             <dbl>
## 1     setosa    50             5.006            3.428             1.462
## 2 versicolor    50             5.936            2.770             4.260
## 3  virginica    50             6.588            2.974             5.552
## # ... with 1 more variables: Petal.Width_mean <dbl>
```

```
## End solution


## * Given the dataset `movies` in package `ggplot2movies`:
```

```
data(movies)
movies
```

```
## # A tibble: 58,788 x 24
##                         title  year length budget rating votes    r1    r2
##                         <chr> <int>  <int>  <int>  <dbl> <int> <dbl> <dbl>
##  1                        $  1971    121     NA    6.4   348   4.5   4.5
##  2        $1000 a Touchdown  1939     71     NA    6.0    20   0.0  14.5
##  3   $21 a Day Once a Month  1941      7     NA    8.2     5   0.0   0.0
##  4                  $40,000  1996     70     NA    8.2     6  14.5   0.0
##  5 $50,000 Climax Show, The  1975     71     NA    3.4    17  24.5   4.5
##  6                    $pent  2000     91     NA    4.3    45   4.5   4.5
##  7                  $windle  2002     93     NA    5.3   200   4.5   0.0
##  8                     '15'  2002     25     NA    6.7    24   4.5   4.5
##  9                      '38  1987     97     NA    6.6    18   4.5   4.5
## 10                  '49-'17  1917     61     NA    6.0    51   4.5   0.0
## # ... with 58,778 more rows, and 16 more variables: r3 <dbl>, r4 <dbl>,
## #   r5 <dbl>, r6 <dbl>, r7 <dbl>, r8 <dbl>, r9 <dbl>, r10 <dbl>,
## #   mpaa <chr>, Action <int>, Animation <int>, Comedy <int>, Drama <int>,
## #   Documentary <int>, Romance <int>, Short <int>
```

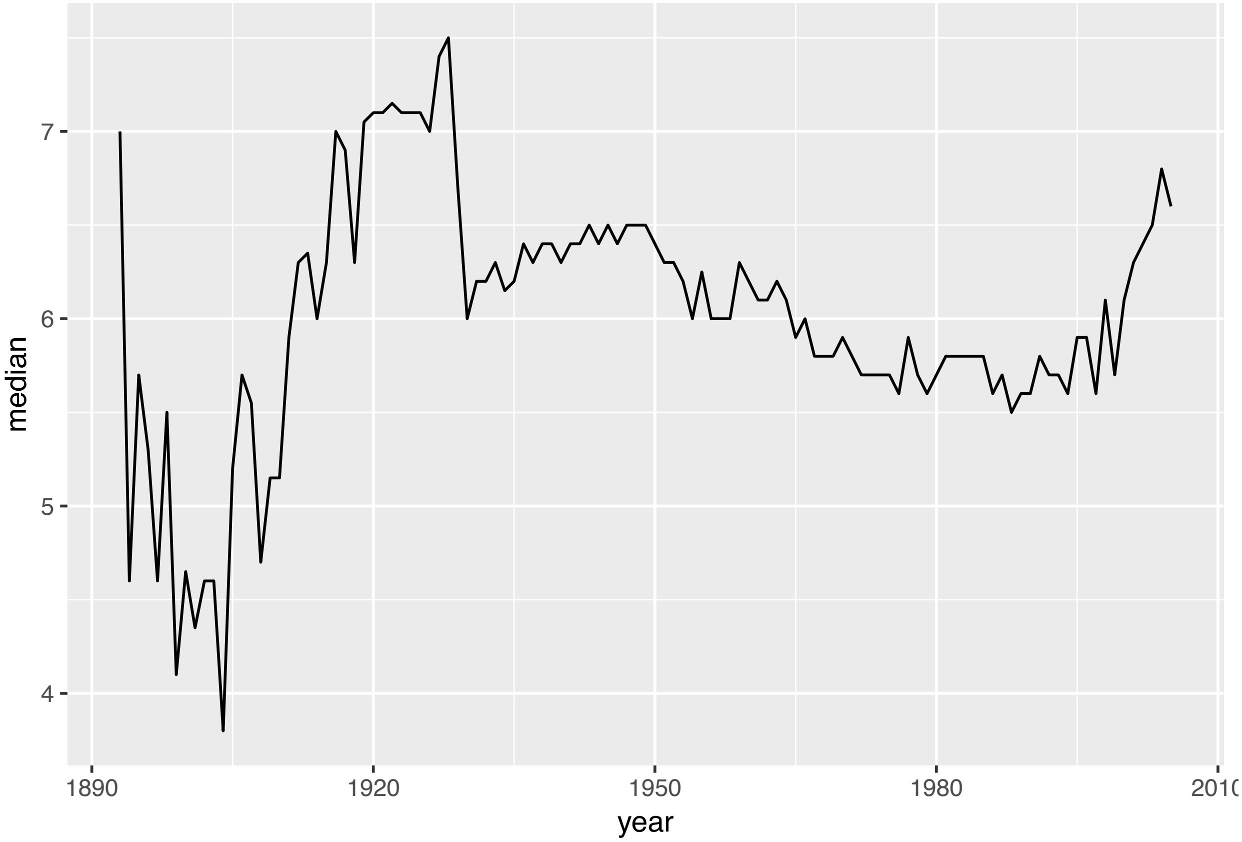
```
## * Get the subset of movies that have a `budget`:
##     1. keeping only columns `title`, `year`, and `budget`
##     2. keeping all columns but `title`, `year`, and `budget`


## Begin solution:
#1
movies %>%
  subset(!is.na(budget)) %>%
  select(title, year, budget)
```

```
## # A tibble: 5,215 x 3
##                          title  year   budget
##                          <chr> <int>    <int>
##  1                    'G' Men  1935   450000
##  2  'Manos' the Hands of Fate  1966    19000
##  3         'Til There Was You  1997 23000000
##  4            .com for Murder  2002  5000000
##  5 10 Things I Hate About You  1999 16000000
##  6              100 Mile Rule  2002  1100000
##  7                  100 Proof  1997   140000
##  8                        101  1989   200000
##  9           101-vy kilometer  2001   200000
## 10             102 Dalmatians  2000 85000000
## # ... with 5,205 more rows
```

```
#2
movies %>%
  subset(!is.na(budget)) %>%
  select(-c(title, year, budget))
```

```
## # A tibble: 5,215 x 21
##    length rating votes    r1    r2    r3    r4    r5    r6    r7    r8
##     <int>  <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     85    7.2   281   0.0   4.5   4.5   4.5   4.5  14.5  34.5  34.5
## 2     74    1.6  7996  74.5   4.5   4.5   4.5   4.5   4.5   4.5   4.5
## 3    113    4.8   799   4.5   4.5   4.5  14.5  14.5  14.5  14.5   4.5
## 4     96    3.7   271  64.5   4.5   4.5   4.5   4.5   4.5   4.5   4.5
## 5     97    6.7 19095   4.5   4.5   4.5   4.5   4.5  14.5  24.5  14.5
## 6     98    5.6   181   4.5   4.5   4.5   4.5  14.5  24.5  14.5  14.5
## 7     94    3.3    19  14.5  14.5   4.5  14.5  14.5  14.5  14.5   0.0
## 8    117    7.8   299   4.5   0.0   4.5   4.5   4.5   4.5   4.5  14.5
## 9    103    5.8     7   0.0   0.0  14.5   0.0   0.0  44.5   0.0  14.5
## 10   100    4.7  1987   4.5   4.5  14.5  14.5  24.5  14.5  14.5   4.5
## # ... with 5,205 more rows, and 10 more variables: r9 <dbl>, r10 <dbl>,
## #   mpaa <chr>, Action <int>, Animation <int>, Comedy <int>, Drama <int>,
## #   Documentary <int>, Romance <int>, Short <int>
```

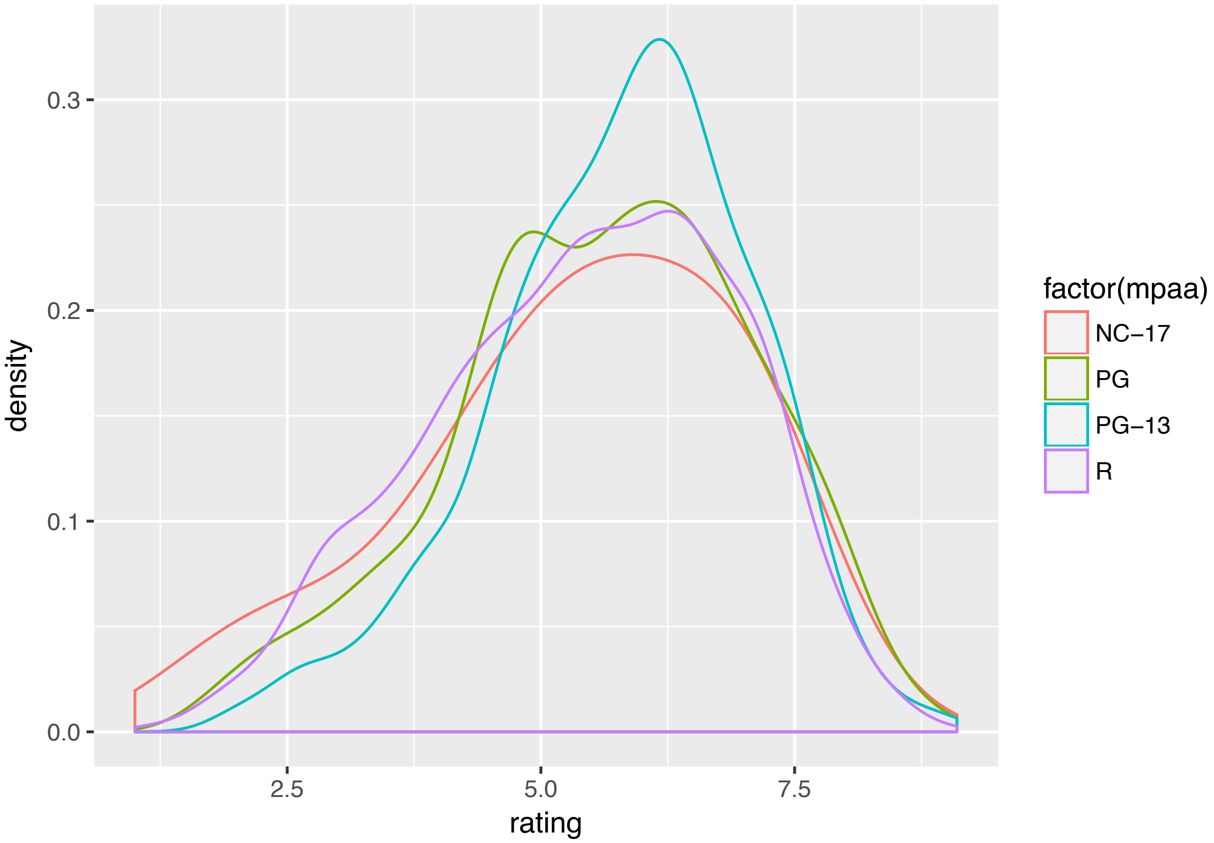
```
## End solution


##  Find median rating per year and plot using ggplot.


## Begin solution
movies %>%
  group_by(year) %>%
  summarise(median = median(rating), na.rm = TRUE) %>%
  ggplot() +
  aes(x = year, y = median) +
  geom_line()
```



```
## End solution

## * For rated movies (`mpaa`):
##     1. Find proportion of rated movies. What do you think of the result?
##     2. Of the rated movies, find distribution (proportion)
##        of ratings. Plot with ggplot.
##     3. Interpret if the distribution has probabilitic meaning or not.


## Begin solution
#1
movies %>%
  subset(mpaa != "") %>%
  summarise(proportion = length(mpaa)/nrow(movies)*100)
```

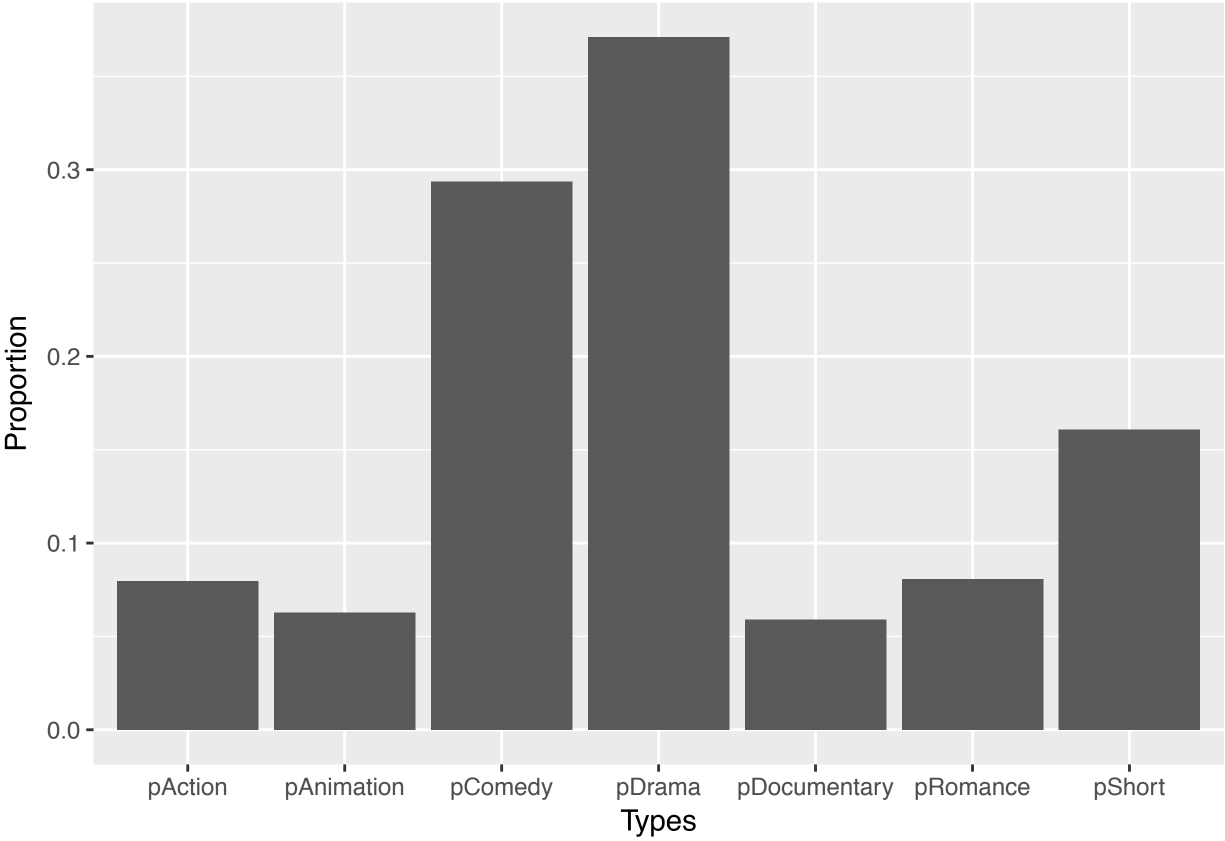
```
## # A tibble: 1 x 1
##   proportion
##        <dbl>
## 1   8.375859
```

```
#The mpaa only includes NC-17, PG, PG-13 and R.
#However, the proportion of rated movies is only 8.376% in all movies.
#I think it is probably because other unrated movies are rated as G(General Audiences).

#2
movies %>%
  group_by(mpaa) %>%
  subset(mpaa != "") %>%
  ggplot() +
  aes(x = rating , colour = factor(mpaa))+
  geom_density()
```

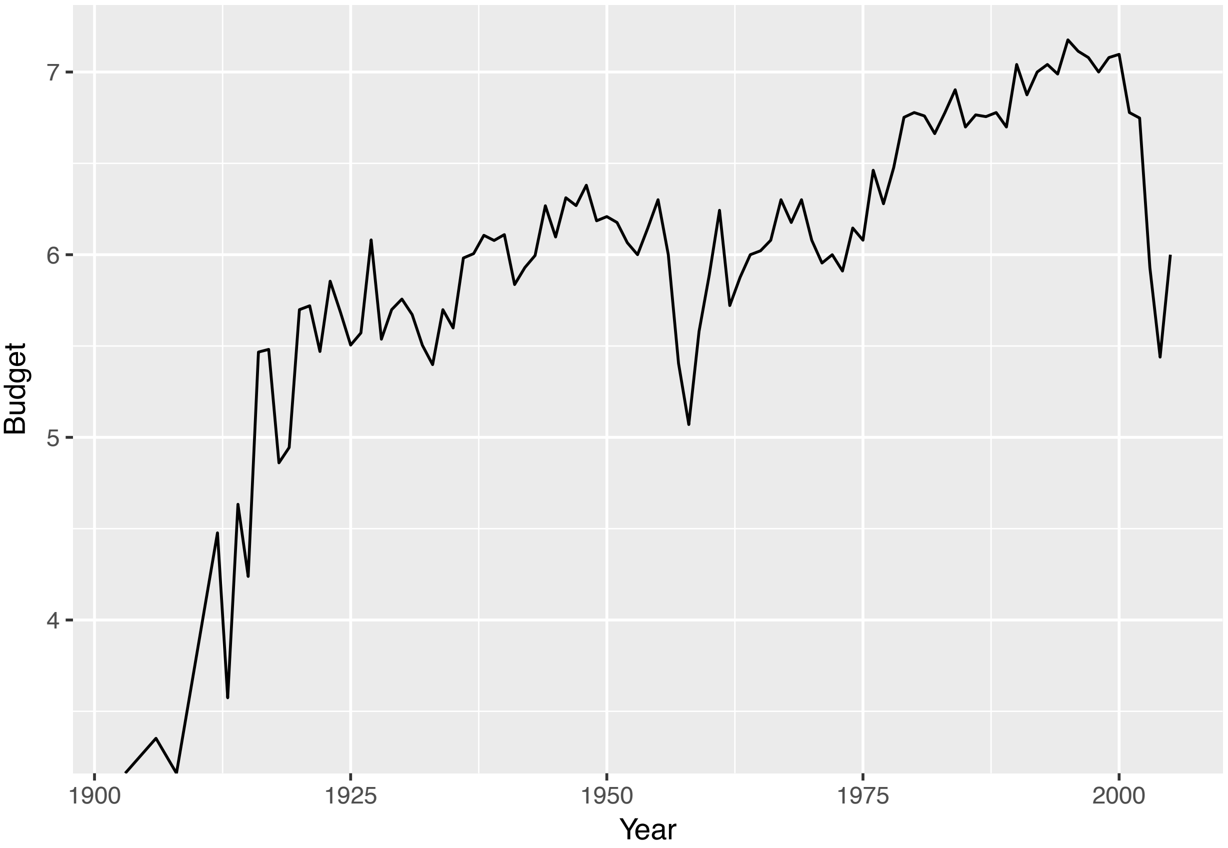


```
#3
#The average rating of all kinds of rated movies are almost the same.
#It's nearly between 5 and 7.

## End solution


##     1. Find the distribution (proportion) of movie types
##     (`"Action"` to `"Short"`). Plot with ggplot.
##     2. Interpret if the distribution has probabilitic meaning or not.


## Begin solution
#1
movies %>%
  summarise(pAction = sum(Action)/nrow(movies),
            pAnimation = sum(Animation)/nrow(movies),
            pComedy = sum(Comedy)/nrow(movies),
            pDrama = sum(Drama)/nrow(movies),
            pDocumentary = sum(Documentary)/nrow(movies),
            pRomance = sum(Romance)/nrow(movies),
            pShort = sum(Short)/nrow(movies)) %>%
  as.data.frame() %>%
  melt() %>%
  ggplot() +
  aes(x = variable, y = value) +
  geom_bar(stat="identity") +
  labs(
    x = "Types",
    y = "Proportion"
    )
```

```
## No id variables; using all as measure variables
```



```
#2
#Because one movie could be more than one type.
#As the result, the Drama and Comedy have more proportion than others.
#I think it's probably beacause people love those two types more.

## End solution


##  Plot yearly $\log_{10}$ median budget with ggplot.

## Begin solution
movies %>%
  subset(!is.na(budget))%>%
  group_by(year) %>%
  summarise(median = log10(median(budget)), na.rm = TRUE) %>%
  ggplot() +
  aes(x = year, y = median) +
  geom_line() +
  labs(
    x = "Year",
    y = "Budget"
  )
```



```
#
## End solution
```