w02

Yekta Amirkhalili

May 2nd, 2023

Week 2 (Session 1) - May 15, 2023

material to cover:

---

0. . . . Data. . .

```
library(rlang)
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
```

```
data <- read.csv('movies.csv')


subData <- select(data, title, year)

head(subData, 5)
```

```
##                       title year
## 1 The Shawshank Redemption 1994
## 2            The Godfather 1972
## 3          The Dark Knight 2008
## 4    The Godfather Part II 1974
## 5             12 Angry Men 1957
```

1. . . . Measure of location, (mean, median and mode). . .

MEAN

```
print('Want to know What is the average rating for movies made in a particular year?')
```

```
## [1] "Want to know What is the average rating for movies made in a particular year?"
```

```
msg = ' Enter the Year: '

yr <- readline(prompt = msg)
```

```
##  Enter the Year:
```

```
yr <- as.integer(yr)

calc_mean <- function(year_){
    particular_year <- subset(data$imbd_rating, data$year == year_)

    rt_avg <- mean(particular_year)
    out <- list(particular_year, rt_avg)

    return(out)
}

func_out <- calc_mean(yr)
output_list <- func_out[1]
output_avg <- func_out[2]

print('Here are some of the ratings from this year: ')
```

```
## [1] "Here are some of the ratings from this year: "
```

```
output_list
```

```
## [[1]]
## numeric(0)
```

```
yr_str <- as.character(yr)

sprintf('The average ratings of the movies on this list from the year %s is %f .', yr_str, output_avg)
```

```
## [1] "The average ratings of the movies on this list from the year NA is NaN ."
```

MEDIAN

```
#same function, except calculate median
calc_median <- function(year_){
    particular_year <- subset(data$imbd_rating, data$year == year_)

    rt_med <- median(particular_year)
```

```
    return(rt_med)
}

median_out <- calc_median(yr)

sprintf('The Median ratings for year %s is %f .', yr_str, median_out)
```

```
## [1] "The Median ratings for year NA is NA ."
```

MODE

```
#same function, except calculate mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

calc_mode <- function(year_){
    particular_year <- subset(data$imbd_rating, data$year == year_)

    rt_mod <- getmode(particular_year)


    return(rt_mod)
}

mode_out <- calc_mode(yr)

sprintf('The Mode ratings for year %s is %f .', yr_str, mode_out)
```

```
## [1] "The Mode ratings for year NA is NA ."
```

...Measure of variability, (variance, standard deviation)...

```
calc_var <- function(year_){
    particular_year <- subset(data$imbd_rating, data$year == year_)

    rt_var <- var(particular_year)


    return(rt_var)
}


sprintf('The Variance of ratings for year %s is %f .', yr_str, calc_var(yr))
```

```
## [1] "The Variance of ratings for year NA is NA ."
```

2. ~~Bayes Rule~~

Introduction

1. Bayes First Rule:

$$P(E) = \sum_{i=1}^{n} P(E|F_i)P(F_i)$$

Ex1. Suppose we have two baskets. Basket A has 3 Red and 2 White balls in it. Basket B has 3 Red and 4 White balls in it. We randomly pick one ball from Basket A and transfer it to Basket B. Then we pick a ball, randomly, out of Basket B. What is the probability of the chosen ball being white?

```
bas_A = c('r','r','r', 'w','w')
bas_B = c('r','r','r', 'w','w','w','w')

pick_white_prob <- function(basket){
    #probability of picking white
    total <- length(basket)
    basket_counts <- table(basket)
    whites <- basket_counts[names(basket_counts) == 'w']

    prob <- whites/total
    return(prob)

}

pick_red_prob <- function(basket){
    #probability of picking white
    total <- length(basket)
    basket_counts <- table(basket)
    reds <- basket_counts[names(basket_counts) == 'r']

    prob <- reds/total
    return(prob)

}
```

- Scenario 1 We pick 1 white ball from Basket A, transfer it to Basket B. Therefore, we now have 3 Red and 5 Whites in Basket B.
- Scenario 2 We pick 1 red ball from Basket A, transfer it to Basket B. Therefore, we now have 4 Red and 4 Whites in Basket B.

In any case, the final probability calculation depends on what happened in the first pick. Let's see this in action!

```
scenario <- function(num){
    #scenario 1
    if(num == 1){
        new_b <- c('r','r','r','w','w','w','w', 'w')
    }else{ #scenario 2
        new_b <- c('r','r','r','r','w','w','w','w')
    }
```

```
    return(new_b)

}
```

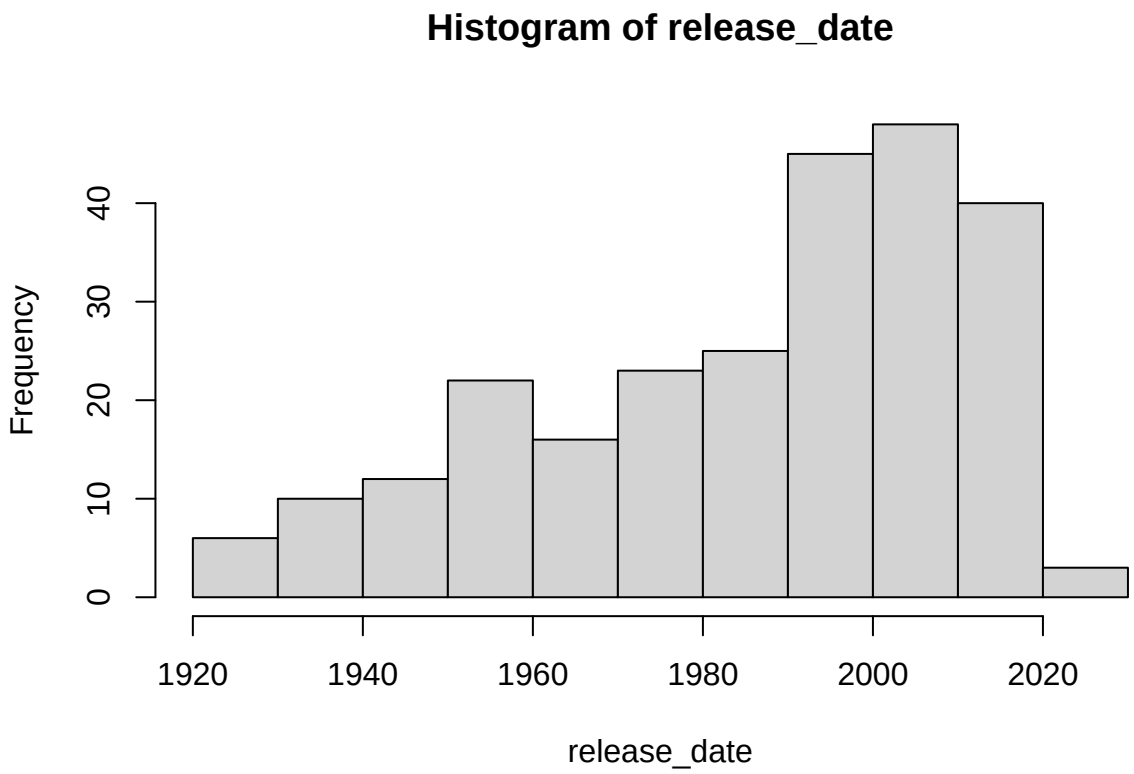
```
newBasket1 <- scenario(1)
newBasket2 <- scenario(2)


prob_ <- function(a, b1, b2){

    # pick white from A, pick white from B of Scenario 1
    total_prob_p1 <- pick_white_prob(a) * pick_white_prob(b1)

    #pick red from A, pick white from new B
    total_prob_p2 <- pick_red_prob(a) * pick_white_prob(b2)

    probability <- total_prob_p1 + total_prob_p2

    return(probability)

}

prob_(bas_A, newBasket1, newBasket2)
```

```
##    w
## 0.55
```

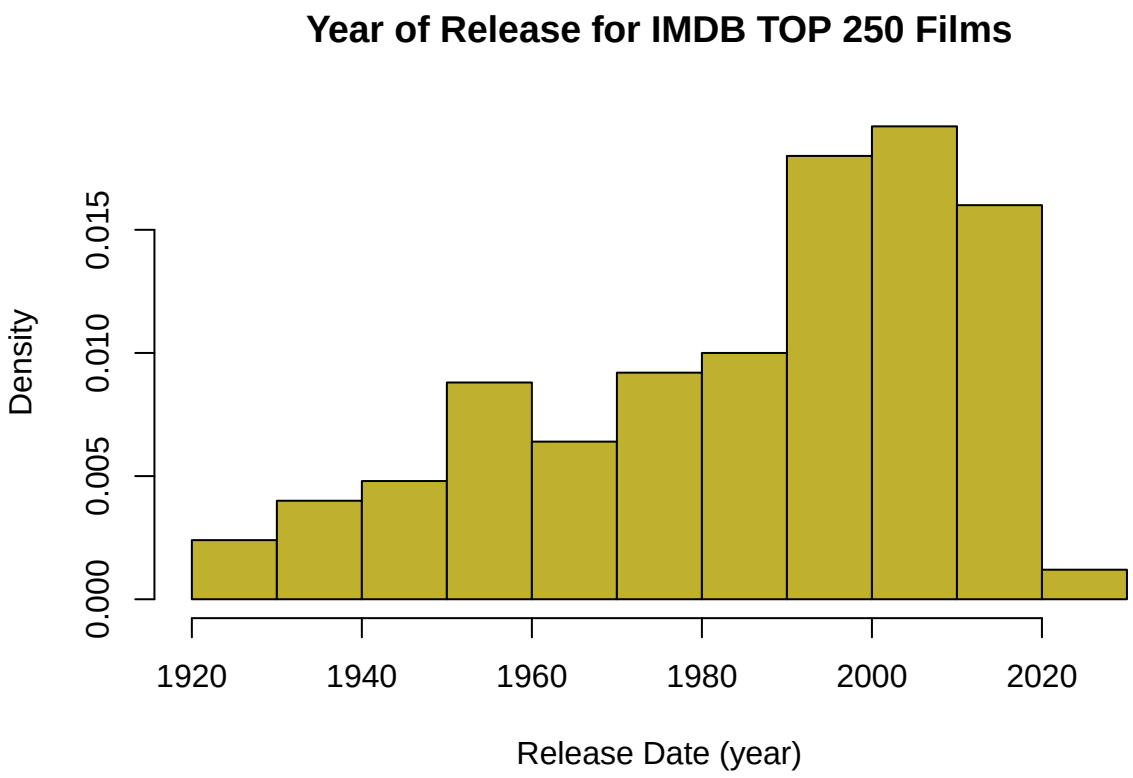
add . . . Histograms. . .

Let's see the distribution of films in the Top 250 based on year of release.

```
release_date <- data$year
hist(release_date)
```




```
hist(release_date,
main="Year of Release for IMDB TOP 250 Films",
xlab="Release Date (year)",
col="#bfb12e",
freq=FALSE)
```



Save your Histogram in a file:

```
# Give the chart file a name.
png(file = "session1_histogram.png")

hist(release_date,
main="Year of Release for IMDB TOP 250 Films",
xlab="Release Date (year)",
col="#bfb12e")

# Save the file.
dev.off()
```

```
## pdf
##   2
```