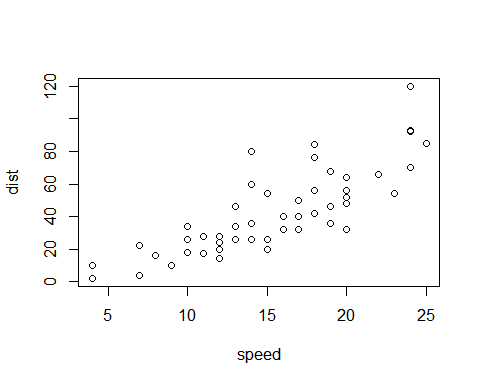
R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

plot(cars)



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

#—————————————-

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(recommenderlab)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loading required package: arules

##   
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

## Loading required package: proxy

##   
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':  
##   
## as.matrix

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

## Loading required package: registry

## Registered S3 methods overwritten by 'registry':  
## method from   
## print.registry\_field proxy  
## print.registry\_entry proxy

polls <- readr::read\_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-04-14/polls.csv')

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## rank = col\_double(),  
## title = col\_character(),  
## artist = col\_character(),  
## gender = col\_character(),  
## year = col\_double(),  
## critic\_name = col\_character(),  
## critic\_rols = col\_character(),  
## critic\_country = col\_character(),  
## critic\_country2 = col\_character()  
## )

rankings <- readr::read\_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-04-14/rankings.csv')

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## ID = col\_double(),  
## title = col\_character(),  
## artist = col\_character(),  
## year = col\_double(),  
## gender = col\_character(),  
## points = col\_double(),  
## n = col\_double(),  
## n1 = col\_double(),  
## n2 = col\_double(),  
## n3 = col\_double(),  
## n4 = col\_double(),  
## n5 = col\_double()  
## )

polls

## # A tibble: 535 x 9  
## rank title artist gender year critic\_name critic\_rols critic\_country  
## <dbl> <chr> <chr> <chr> <dbl> <chr> <chr> <chr>   
## 1 1 Term~ Publi~ male 1998 Joseph Aba~ Fat Beats US   
## 2 2 4th ~ Gza f~ male 1995 Joseph Aba~ Fat Beats US   
## 3 3 Pete~ Run D~ male 1986 Joseph Aba~ Fat Beats US   
## 4 4 Play~ GLOBE~ male 2001 Joseph Aba~ Fat Beats US   
## 5 5 Time~ O.C. male 1994 Joseph Aba~ Fat Beats US   
## 6 1 Play~ Slum ~ male 1997 Biba Adams Critic US   
## 7 2 Self~ Stop ~ mixed 1989 Biba Adams Critic US   
## 8 3 Push~ Salt-~ female 1986 Biba Adams Critic US   
## 9 4 Ambi~ 2Pac male 1996 Biba Adams Critic US   
## 10 5 Big ~ JAY-Z~ male 1999 Biba Adams Critic US   
## # ... with 525 more rows, and 1 more variable: critic\_country2 <chr>

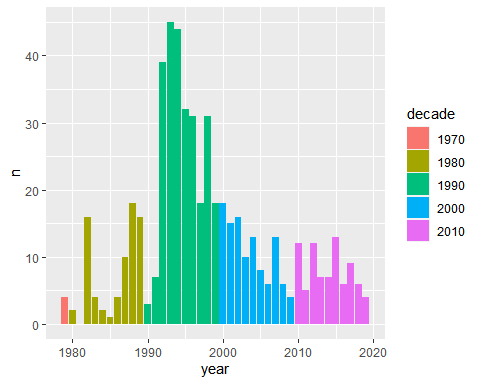
rankings

## # A tibble: 311 x 12  
## ID title artist year gender points n n1 n2 n3 n4 n5  
## <dbl> <chr> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 Juicy The No~ 1994 male 140 18 9 3 3 1 2  
## 2 2 Fight ~ Public~ 1989 male 100 11 7 3 1 0 0  
## 3 3 Shook ~ Mobb D~ 1995 male 94 13 4 5 1 1 2  
## 4 4 The Me~ Grandm~ 1982 male 90 14 5 3 1 0 5  
## 5 5 Nuthin~ Dr Dre~ 1992 male 84 14 2 4 2 4 2  
## 6 6 C.R.E.~ Wu-Tan~ 1993 male 62 10 3 1 1 4 1  
## 7 7 93 ’Ti~ Souls ~ 1993 male 50 7 2 2 2 0 1  
## 8 8 Passin~ The Ph~ 1992 male 48 6 3 2 0 0 1  
## 9 9 N.Y. S~ Nas 1994 male 46 7 1 3 1 1 1  
## 10 10 Dear M~ 2Pac 1995 male 42 6 2 1 1 2 0  
## # ... with 301 more rows

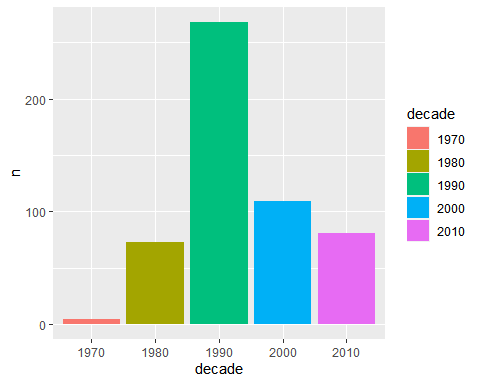
polls %>%   
 count(title, sort = TRUE)

## # A tibble: 309 x 2  
## title n  
## <chr> <int>  
## 1 Juicy 18  
## 2 Nuthin’ But A ‘G’ Thang 14  
## 3 The Message 14  
## 4 Shook Ones (Part II) 13  
## 5 Fight The Power 11  
## 6 C.R.E.A.M. 10  
## 7 93 ’Til Infinity 7  
## 8 N.Y. State Of Mind 7  
## 9 Dear Mama 6  
## 10 Jesus Walks 6  
## # ... with 299 more rows

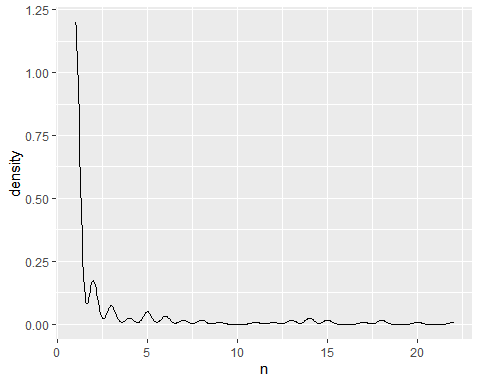
#getting golden year  
polls %>%   
 count(year) %>%   
 mutate(decade = floor(year / 10)\*10) %>%   
 mutate(decade = as.factor(decade)) %>%   
 ggplot(aes(x = year, y = n, fill = decade)) + geom\_col()



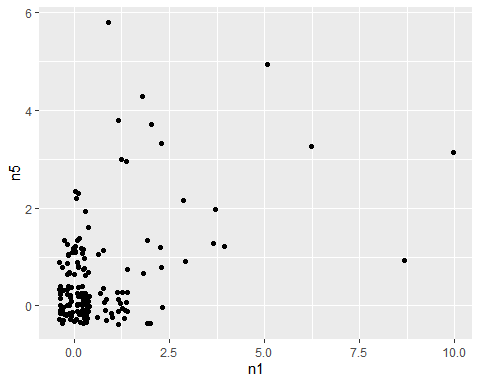
#The golden-age of rap is the 1990s   
polls %>%   
 count(year) %>%   
 mutate(decade = floor(year / 10)\*10) %>%   
 mutate(decade = as.factor(decade)) %>%   
 ggplot(aes(x = decade, y = n, fill = decade)) + geom\_col()



#great artist  
polls %>%   
 count(artist, sort = TRUE) %>%   
 ggplot(aes(x = n)) + geom\_density()



#artist plot  
rankings %>%   
 select(artist, n, n1, n2, n3, n4, n5) %>%   
 group\_by(artist) %>%   
 summarise\_all(sum) %>%   
 filter(!str\_detect(artist, "ft.")) %>%   
 ggplot(aes(x = n1, y = n5)) + geom\_jitter()



#overview  
rankings %>%   
 select(artist, n, n1, n2, n3, n4, n5) %>%   
 group\_by(artist) %>%   
 summarise\_all(sum) %>%   
 filter(!str\_detect(artist, "ft.")) %>%   
 arrange(desc(n1)) %>%   
 slice(1:5)

## # A tibble: 5 x 7  
## artist n n1 n2 n3 n4 n5  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 The Notorious B.I.G. 22 10 3 4 2 3  
## 2 Public Enemy 18 9 4 2 2 1  
## 3 Wu-Tang Clan 20 6 1 3 7 3  
## 4 Grandmaster Flash & The Furious Five 14 5 3 1 0 5  
## 5 2Pac 13 4 3 1 4 1

rankings %>%   
 select(artist, n, n1, n2, n3, n4, n5) %>%   
 group\_by(artist) %>%   
 summarise\_all(sum) %>%   
 filter(!str\_detect(artist, "ft.")) %>%   
 arrange(desc(n5)) %>%   
 slice(1:5)

## # A tibble: 5 x 7  
## artist n n1 n2 n3 n4 n5  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Kendrick Lamar 15 1 3 4 1 6  
## 2 Grandmaster Flash & The Furious Five 14 5 3 1 0 5  
## 3 JAY-Z 18 2 5 3 4 4  
## 4 Kanye West 14 1 1 4 4 4  
## 5 OutKast 17 2 4 3 4 4

#15% of the top songs were voted by only one country   
#Home town / country bias?   
polls %>%   
 count(title, critic\_country, name = "song\_nom") %>%   
 add\_count(title, name = "number\_of\_countries") %>%   
 filter(number\_of\_countries == 1 & critic\_country != "US") %>%   
 nrow() / nrow(polls)

## [1] 0.1551402

polls %>%   
 count(title, critic\_country, name = "song\_nom\_country") %>%   
 add\_count(title, name = "number\_of\_countries") %>%   
 filter(number\_of\_countries != 1) %>%   
 select(-number\_of\_countries) %>%   
 pivot\_wider(names\_from = "critic\_country", values\_from = "song\_nom\_country")

## # A tibble: 52 x 14  
## title `South Africa` US Germany India Japan UK `New Zealand`  
## <chr> <int> <int> <int> <int> <int> <int> <int>  
## 1 93 ’~ 1 6 NA NA NA NA NA  
## 2 All ~ NA 2 1 NA NA NA NA  
## 3 Alri~ NA 4 NA 1 NA NA NA  
## 4 B.O.~ NA 3 NA NA 1 NA NA  
## 5 Bloo~ NA NA 1 NA NA 1 NA  
## 6 Boda~ NA 1 NA NA NA NA 1  
## 7 Brin~ NA 1 NA 1 NA NA NA  
## 8 C.R.~ NA 7 NA NA NA 1 1  
## 9 Cali~ NA 1 1 NA NA NA NA  
## 10 Can’~ NA 1 1 NA NA NA NA  
## # ... with 42 more rows, and 6 more variables: `Russian Federation` <int>,  
## # China <int>, Nigeria <int>, Kenya <int>, Canada <int>, Colombia <int>

library(recommenderlab)

rap\_matrix <- polls %>%   
 select(critic\_name, title) %>%   
 mutate(n = 1) %>%   
 arrange(title) %>%   
 pivot\_wider(names\_from = "title", values\_from = "n", values\_fill = list(n = 0)) %>%   
 select(-critic\_name) %>%   
 as.matrix() %>%   
 as("binaryRatingMatrix")

training\_schema <- evaluationScheme(rap\_matrix, method = "split", train = .8, given = -1)  
training\_schema

## Evaluation scheme using all-but-1 items  
## Method: 'split' with 1 run(s).  
## Training set proportion: 0.800  
## Good ratings: NA  
## Data set: 107 x 309 rating matrix of class 'binaryRatingMatrix' with 535 ratings.

UBCF\_Model <- evaluate(training\_schema, method = "UBCF", type = "topNList", n = 5)

## UBCF run fold/sample [model time/prediction time]  
## 1 [0.02sec/0.35sec]

IBCF\_Model <- evaluate(training\_schema, method = "IBCF", type = "topNList", n = 5)

## IBCF run fold/sample [model time/prediction time]  
## 1 [0.49sec/0.17sec]

UBCF\_Model %>% avg()

## TP FP FN TN precision recall TPR FPR  
## 5 0 4.454545 1 304.5455 0 0 0 0.014416

IBCF\_Model %>% avg() %>% as\_tibble()

## # A tibble: 1 x 8  
## TP FP FN TN precision recall TPR FPR  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0455 4.95 0.955 304. 0.00909 0.0455 0.0455 0.0160

tune\_engines <- function(schema, parameters){  
   
 UBCF\_Model <- evaluate(schema, method = "UBCF", type = "topNList", n = 5, param = list(nn = parameters))  
 IBCF\_Model <- evaluate(schema, method = "IBCF", type = "topNList", n = 5, param = list(k = parameters))  
   
   
 UBCF\_Model %>%   
 avg() %>%   
 as\_tibble() %>%   
 mutate(model = "UBCF") %>%   
 rbind(IBCF\_Model %>%   
 avg() %>%   
 as\_tibble() %>%   
 mutate(model = "IBCF")) %>%   
 return()  
   
   
  
}

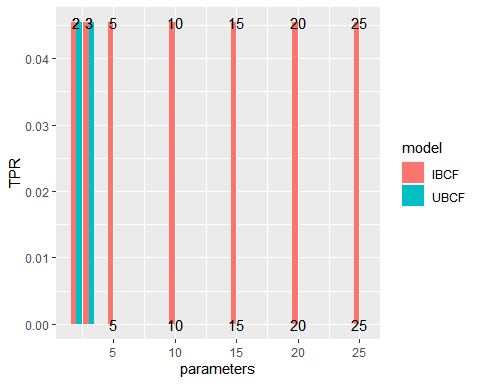
tune\_grid <- tibble(parameters = c(2, 3, 5, 10, 15, 20, 25))  
  
  
history <- tune\_grid %>%   
 mutate(results = map(parameters, ~tune\_engines(training\_schema, .x))) %>%   
 unnest()

## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.18sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.07sec/0sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.15sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.06sec/0.02sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.2sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.08sec/0.02sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.16sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.06sec/0.01sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.14sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.06sec/0.02sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.16sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.06sec/0.03sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.16sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.14sec/0.02sec]

## Warning: `cols` is now required when using unnest().  
## Please use `cols = c(results)`

#taking neighbours

#Use 5 nearest neighbros  
history %>%   
 ggplot(aes(x = parameters, y = TPR, fill = model, label = parameters)) + geom\_col(position = "dodge") + geom\_text(aes(x = parameters, y = TPR))



UBCF\_Final\_model <- Recommender(getData(training\_schema, "train"), "UBCF", param = list(nn = 5))  
  
  
UBCF\_Final\_model

## Recommender of type 'UBCF' for 'binaryRatingMatrix'   
## learned using 85 users.

#predictions

predictions <- predict(UBCF\_Final\_model, getData(training\_schema, "known"), type = "topNList")  
calcPredictionAccuracy(predictions, getData(training\_schema,"unknown"), given = -1)

## TP FP FN TN precision recall   
## 0.04545455 8.22727273 0.95454545 300.77272727 0.00500000 0.04545455   
## TPR FPR   
## 0.04545455 0.02662548

#engine ready now ..

rec\_engine <- Recommender(rap\_matrix, "UBCF", param = list(nn = 5))  
rec\_engine

## Recommender of type 'UBCF' for 'binaryRatingMatrix'   
## learned using 107 users.

polls %>% filter(str\_detect(artist, "2Pac")) %>% distinct(title) %>% arrange(title)

## # A tibble: 8 x 1  
## title   
## <chr>   
## 1 Ambitionz Az A Ridah  
## 2 Brenda’s Got A Baby   
## 3 California Love   
## 4 Changes   
## 5 Dear Mama   
## 6 Hit ’Em Up   
## 7 I Get Around   
## 8 So Many Tears

#my seclections

akash\_songs <- polls %>%   
 select(title) %>%   
 distinct() %>%   
 arrange(title) %>%   
 filter(title %in% c("All Of The Lights", "Alright", "Bitch Don’t Kill My Vibe", "m.A.A.d. city", "Changes")) %>%   
 rbind(polls %>% select(title) %>% distinct()) %>%   
 count(title) %>%   
 mutate(n = n -1) %>%   
 pivot\_wider(names\_from = "title", values\_from = "n", values\_fill = list(n = 0)) %>%   
 as.matrix() %>%   
 as("binaryRatingMatrix")

rec\_engine

## Recommender of type 'UBCF' for 'binaryRatingMatrix'   
## learned using 107 users.

#final predictions

predict(rec\_engine, akash\_songs) %>% as("list") %>% as.data.frame()

## X1  
## 1 1 Train  
## 2 Black Steel In The Hour Of Chaos  
## 3 Bodak Yellow  
## 4 Country Grammar  
## 5 Da Mystery Of Chessboxin'  
## 6 Day 'N' Nite  
## 7 Doo Wop (That Thing)  
## 8 Double Trouble At The Amphitheatre  
## 9 Get By  
## 10 Get Ur Freak On