Tutorial 2

Priority

```
2 * 1:5 # : takes precedence over *
```

```
## [1]  2  4  6  8 10
```

```
1: 5^2 # ^ takes precedence over :
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

':' takes precedence over '*'.

Matrix Formation

The number of rows and columns do not have to be exact

```
matrix(10:15, ncol=2, nrow=2) # truncate
```

```
##      [,1] [,2]
## [1,]   10   12
## [2,]   11   13
```

```
matrix(10:15, ncol=3, nrow=3) # recycle
```

```
##      [,1] [,2] [,3]
## [1,]   10   13   10
## [2,]   11   14   11
## [3,]   12   15   12
```

```
# truncate + warning, 7 numbers will not fit in (not a multiple of the rows)
matrix(10:16, ncol=2, nrow=2)
```

```
## Warning in matrix(10:16, ncol = 2, nrow = 2): data length [7] is not a sub-
## multiple or multiple of the number of rows [2]
```

```
##      [,1] [,2]
## [1,]   10   12
## [2,]   11   13
```

```
# will first produce the recycled version (cause warning):
# 10 12 14 16
# 11 13 15 10

# and then truncate
```

NOTE: A warning will always apply when matrix/vector lengths do not match. (e.g. 3:7 + 1:2)

Matrix Selection

```
m = matrix(10:16, ncol=4, nrow=3)
```

```
## Warning in matrix(10:16, ncol = 4, nrow = 3): data length [7] is not a sub-
## multiple or multiple of the number of rows [3]
```

```
m[3,-2]
```

```
## [1] 12 11 14
```

```
# 10 13 16 12
# 11 14 10 13
# 12 15 11 14

# Look at row 3, exclude column 2 -> 12 11 14
```

Matrix Naming

```
x = c(A=1, B=2, C=3)
y = c(James=4, John=5, Joe=6)
z = x-y # takes the names of the first vector x
z
```

```
##  A  B  C
## -3 -3 -3
```

Matrix Function

```
sin(1:10) # function applied onto all elements
```

```
##  [1]  0.8414710  0.9092974  0.1411200 -0.7568025 -0.9589243 -0.2794155
##  [7]  0.6569866  0.9893582  0.4121185 -0.5440211
```

Duplication

```
rep(1:5, 2)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5
```

```
c(1:5, 1:5)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5
```

Self-practice

1. Generate 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1

```
rep_len(1:5, 16)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1
```

```
rep(1:5, length.out=16)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1
```

```
matrix(1:5, ncol=16)[1,] # throws warning
```

```
## Warning in matrix(1:5, ncol = 16): data length [5] is not a sub-multiple or
## multiple of the number of columns [16]
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1
```

2. Split the string “Jack Wong, Liu Qizhang, Tingting Koh, Carol Tan” into 4 names and store in a vector.

```
name = "Jack Wong, Liu Qizhang, Tingting Koh, Carol Tan"
strsplit(name, ", ")[[1]]
```

```
## [1] "Jack Wong"    "Liu Qizhang"  "Tingting Koh" "Carol Tan"
```

```
unlist(strsplit(name, ", "))
```

```
## [1] "Jack Wong"    "Liu Qizhang"  "Tingting Koh" "Carol Tan"
```

3.

```
result = c(James="97,A+",Tom="87,A",Jack="50,C",Carol="67,B")

# substring
library(stringr)
loc = str_locate(result,",")[,"start"]
score = as.integer(substring(result, 0, loc - 1))
names(score) = names(result)
score
```

```
## James   Tom  Jack Carol
##    97    87    50    67
```

```
grade = substring(result, loc + 1)
grade
```

```
## James   Tom  Jack Carol
##  "A+"   "A"   "C"   "B"
```

```
#strplit and matrix
m = matrix(unlist(strsplit(result, ",")), nrow=2)
score = as.integer(m[1,])
```

4.

```
salaries = c(Tom=3000, James=7000, Grace= 5000, Wong=3500, Wong=5000, Grace=6000)
dupes = duplicated(names(salaries))
salaries[dupes]
```

```
##  Wong Grace
##  5000  6000
```

5.

```
salaries = c(Tom=3000, James=7000, Grace= 5000, Wong=3500, Wong=5000)
ave = mean(salaries)
names(salaries[salaries > ave])
```

```
## [1] "James" "Grace" "Wong"
```

Tutorial 3

Tutorial 3

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

Missing values

```
head(is.na(airquality)) # all VALUES with NA, some rows have multiple
```

```
##      Ozone Solar.R  Wind  Temp Month   Day
## [1,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [5,]  TRUE    TRUE FALSE FALSE FALSE FALSE
## [6,] FALSE    TRUE FALSE FALSE FALSE FALSE
```

```
head(complete.cases(airquality)) # which ROWS are complete
```

```
## [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE
```

```
# No. of incomplete records
sum(!complete.cases(airquality))
```

```
## [1] 42
```

Clean Data

```
# method 1
airquality.clean = airquality[complete.cases(airquality), ]

# method 2
library(tidyr)
airquality.clean = drop_na(airquality)

# method 3
airquality.clean = na.omit(airquality)
```

Subset

```
# method 1
subset = airquality.clean[airquality.clean$Month==6, ]
dim(subset)
```

```
## [1] 9 6
```

```
# method 2
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
```

```
airquality.clean %>% filter(Month==6) %>% dim()
```

```
## [1] 9 6
```

```
# method 3
subset(airquality.clean, Month==6)
```

```
##    Ozone Solar.R Wind Temp Month Day
## 38    29     127  9.7   82     6   7
## 40    71     291 13.8   90     6   9
## 41    39     323 11.5   87     6  10
## 44    23     148  8.0   82     6  13
## 47    21     191 14.9   77     6  16
## 48    37     284 20.7   72     6  17
## 49    20      37  9.2   65     6  18
## 50    12     120 11.5   73     6  19
## 51    13     137 10.3   76     6  20
```

Efficient Counting

```
# method 1
# creates subset -> slow
subset = airquality.clean[airquality.clean$Wind >=7 & airquality.clean$Wind <= 8, ]
nrow(subset)
```

```
## [1] 16
```

```
# method 2
# not storing subset -> faster
sum(airquality.clean$Wind >=7 & airquality.clean$Wind <= 8)
```

```
## [1] 16
```

```
library(dplyr)
airquality.clean = airquality.clean %>% mutate(Index=Solar.R*Wind/Temp)
airquality.clean[5, "Index"]
```

```
## [1] 39.56
```

```
head(airquality.clean[5, "Index"])
```

```
## [1] 39.56
```

Last Day of the Month

```
# method 1
airquality.clean %>% group_by(Month) %>% filter(Day == max(Day))
```

```
## # A tibble: 5 x 7
## # Groups:   Month [5]
##   Ozone Solar.R  Wind  Temp Month   Day Index
##   <int>   <int> <dbl> <int> <int> <int> <dbl>
## 1    37     279   7.4    76     5    31  27.2
## 2    13     137  10.3    76     6    20  18.6
## 3    59     254   9.2    81     7    31  28.8
## 4    85     188   6.3    94     8    31  12.6
## 5    20     223  11.5    68     9    30  37.7
```

```
# method 2
max.month.day = aggregate(airquality.clean$Day, by=list(airquality.clean$Month), max)
# then use join
merge(airquality.clean, max.month.day, by.x=c("Month", "Day"), by.y=c("Group.1", "x"))
```

```
##   Month Day Ozone Solar.R Wind Temp    Index
## 1     5  31    37     279  7.4   76 27.16579
## 2     6  20    13     137 10.3   76 18.56711
## 3     7  31    59     254  9.2   81 28.84938
## 4     8  31    85     188  6.3   94 12.60000
## 5     9  30    20     223 11.5   68 37.71324
```

```
residents_raw = read.csv("./Data/singapore-residents-by-age-group-ethnic-group-and-sex-end-june-annual.csv")

# We want the columns to be only Year, Gender, Race, Value
```

```
unique(unlist(strsplit(residents_raw$level_1, " ")))
```

```
##  [1] "Total"     "Residents" "Male"      "Female"    "Malays"    "Chinese"
##  [7] "Indians"   "Other"     "Ethnic"    "Groups"    "(Total)"   "(Males)"
## [13] "(Females)"
```

Tutorial 4

1

```
library("jsonlite")
```

```
## Warning: package 'jsonlite' was built under R version 4.1.1
```

```
# yesterday's information
url = "https://api.data.gov.sg/v1/transport/taxi-availability?date_time=2021-08-30T12:00:00"

data = fromJSON(url)

taxi_coords = as.data.frame(data$features$geometry$coordinates)

head(taxi_coords)
```

```
##         X1       X2
## 1 103.6230 1.289270
## 2 103.6257 1.274740
## 3 103.6376 1.300310
## 4 103.6377 1.300390
## 5 103.6553 1.314034
## 6 103.6561 1.305540
```

2

```
library("curl")
```

```
## Warning: package 'curl' was built under R version 4.1.1
```

```
## Using libcurl 7.64.1 with Schannel
```

```
library("XML")
```

```
## Warning: package 'XML' was built under R version 4.1.1
```

```
theurl = "https://www.ncaa.com/rankings/basketball-men/d1/ncaa-mens-basketball-net-rankings"
url = curl(theurl)
urldata = readLines(url)
basketball_data = readHTMLTable(urldata, stringAsFactors=F)

head(basketball_data[[1]])
```

```
##   Rank Previous              School Conference Record Road Neutral Home Quad 1
## 1    1        1             Gonzaga        WCC   31-1  7-0    12-1 12-0   12-1
## 2    2        2              Baylor     Big 12   28-2  7-1    10-1 11-0   13-2
## 3    3        4            Michigan    Big Ten   23-5  6-2     4-2 13-1   10-3
## 4    4        3            Illinois    Big Ten   24-7  9-3     4-2 11-2   12-5
## 5    5        5             Houston        AAC   27-4  5-3     8-1 14-0    6-2
## 6    6       19 Southern California     Pac-12   25-8  7-3     5-3 13-2    9-7
##   Quad 2 Quad 3 Quad 4
## 1    6-0    7-0    6-0
## 2    3-0    7-0    5-0
## 3    4-2    7-0    2-0
## 4    5-2    5-0    2-0
## 5    5-1   13-1    3-0
## 6    7-1    6-0    3-0
```

```
theurl = "https://cloudatlas.wmo.int/en/appendix-1-etymology-of-latin-names-of-clouds.html"
url = curl(theurl)
urldata = readLines(url)
cloud_data = readHTMLTable(urldata, stringAsFactors=F)
species_data = cloud_data[2]

head(species_data)
```

```
## $`NULL`
##              V1
## 1      Fibratus
## 2       Uncinus
## 3     Spissatus
## 4   Castellanus
## 5       Floccus
## 6  Stratiformis
## 7     Nebulosus
## 8  Lenticularis
## 9       Fractus
## 10      Humilis
## 11    Mediocris
## 12    Congestus
## 13       Calvus
## 14   Capillatus
## 15      Volutus
##
V2
## 1
From the Latin fibratus, which means fibrous, possessing fibres, filaments
## 2
From the Latin uncinus, which means hooked
## 3                                                                          From the La
tin spissatus, past participle of the verb spissare, which means to make thick, to condense
## 4                                                                    From the Latin ca
stellanus, derived from castellum, which means a castle or the enceinte of a fortified town
## 5
From the Latin floccus, which means tuft of wool, fluff, nap of cloth
## 6  From the Latin stratus, past participle of the verb sternere, which means to extend, to
spread out, to flatten out, to cover with a layer, and forma, which means form, appearance
## 7
From the Latin nebulosus, which means full of mist, covered with fog, nebulous
## 8
From the Latin lenticularis, derived from lenticula, diminutive of lens meaning a lentil
## 9                                                              From the Latin fractus, p
ast participle of the verb frangere, which means to shatter, to break, to snap, to fracture
## 10
From the Latin humilis, which means near the ground, low, of small size
## 11
From the Latin mediocris, which means medium, keeping to the middle
## 12                                                               From the Latin congestu
s, past participle of the verb congerere, which means to pile up, to heap up, to accumulate
## 13                                                                   From the Latin
calvus, which means bald, and, in a wider sense, is applied to something stripped or bared
## 14
From the Latin capillatus, which means having hair, derived from capillus, which means hair
## 15
From the Latin volutus, which means rolled
```

3

How to get HTML data from a webpage? Use SearchGadget (Chrome extension), rvest and tidyverse

```
library(rvest)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.1
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

```
## Warning: package 'forcats' was built under R version 4.1.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()         masks stats::filter()
## x purrr::flatten()        masks jsonlite::flatten()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x readr::parse_date()     masks curl::parse_date()
```

```
## To obtain links to each NBA team (There are 30 teams in the NBA)
url <- "http://www.espn.com/nba/players"
page <- read_html(url) # reads the whole page
nodes <- html_nodes(page, ".small-logos div a")

length(nodes) # 30 teams
```

```
## [1] 30
```

```
nodes[[1]] # href stores the url
```

```
## {html_node}
## <a style="padding-top:5px;padding-left:0px;" href="/nba/teams/roster?team=Bos">
```

```
# note that the url is incomplete

rosters = html_attr(nodes,"href")
rosters
```

```
##  [1] "/nba/teams/roster?team=Bos"  "/nba/teams/roster?team=BKN"
##  [3] "/nba/teams/roster?team=NY"   "/nba/teams/roster?team=Phi"
##  [5] "/nba/teams/roster?team=Tor"  "/nba/teams/roster?team=GS"
##  [7] "/nba/teams/roster?team=LAC"  "/nba/teams/roster?team=LAL"
##  [9] "/nba/teams/roster?team=PHX"  "/nba/teams/roster?team=Sac"
## [11] "/nba/teams/roster?team=Chi"  "/nba/teams/roster?team=Cle"
## [13] "/nba/teams/roster?team=Det"  "/nba/teams/roster?team=Ind"
## [15] "/nba/teams/roster?team=Mil"  "/nba/teams/roster?team=Dal"
## [17] "/nba/teams/roster?team=Hou"  "/nba/teams/roster?team=Mem"
## [19] "/nba/teams/roster?team=NO"   "/nba/teams/roster?team=SA"
## [21] "/nba/teams/roster?team=Atl"  "/nba/teams/roster?team=Cha"
## [23] "/nba/teams/roster?team=Mia"  "/nba/teams/roster?team=Orl"
## [25] "/nba/teams/roster?team=WSH"  "/nba/teams/roster?team=Den"
## [27] "/nba/teams/roster?team=Min"  "/nba/teams/roster?team=Okc"
## [29] "/nba/teams/roster?team=Por"  "/nba/teams/roster?team=UTAH"
```

```
url_header = "http://www.espn.com"
urls = paste0(url_header, rosters)
urls
```

```
##  [1] "http://www.espn.com/nba/teams/roster?team=Bos"
##  [2] "http://www.espn.com/nba/teams/roster?team=BKN"
##  [3] "http://www.espn.com/nba/teams/roster?team=NY"
##  [4] "http://www.espn.com/nba/teams/roster?team=Phi"
##  [5] "http://www.espn.com/nba/teams/roster?team=Tor"
##  [6] "http://www.espn.com/nba/teams/roster?team=GS"
##  [7] "http://www.espn.com/nba/teams/roster?team=LAC"
##  [8] "http://www.espn.com/nba/teams/roster?team=LAL"
##  [9] "http://www.espn.com/nba/teams/roster?team=PHX"
## [10] "http://www.espn.com/nba/teams/roster?team=Sac"
## [11] "http://www.espn.com/nba/teams/roster?team=Chi"
## [12] "http://www.espn.com/nba/teams/roster?team=Cle"
## [13] "http://www.espn.com/nba/teams/roster?team=Det"
## [14] "http://www.espn.com/nba/teams/roster?team=Ind"
## [15] "http://www.espn.com/nba/teams/roster?team=Mil"
## [16] "http://www.espn.com/nba/teams/roster?team=Dal"
## [17] "http://www.espn.com/nba/teams/roster?team=Hou"
## [18] "http://www.espn.com/nba/teams/roster?team=Mem"
## [19] "http://www.espn.com/nba/teams/roster?team=NO"
## [20] "http://www.espn.com/nba/teams/roster?team=SA"
## [21] "http://www.espn.com/nba/teams/roster?team=Atl"
## [22] "http://www.espn.com/nba/teams/roster?team=Cha"
## [23] "http://www.espn.com/nba/teams/roster?team=Mia"
## [24] "http://www.espn.com/nba/teams/roster?team=Orl"
## [25] "http://www.espn.com/nba/teams/roster?team=WSH"
## [26] "http://www.espn.com/nba/teams/roster?team=Den"
## [27] "http://www.espn.com/nba/teams/roster?team=Min"
## [28] "http://www.espn.com/nba/teams/roster?team=Okc"
## [29] "http://www.espn.com/nba/teams/roster?team=Por"
## [30] "http://www.espn.com/nba/teams/roster?team=UTAH"
```

```
# names are not hidden
teams = html_text(nodes)
head(teams)
```

```
## [1] "Boston Celtics"        "Brooklyn Nets"         "New York Knicks"
## [4] "Philadelphia 76ers"    "Toronto Raptors"       "Golden State Warriors"
```

Tutorial 5

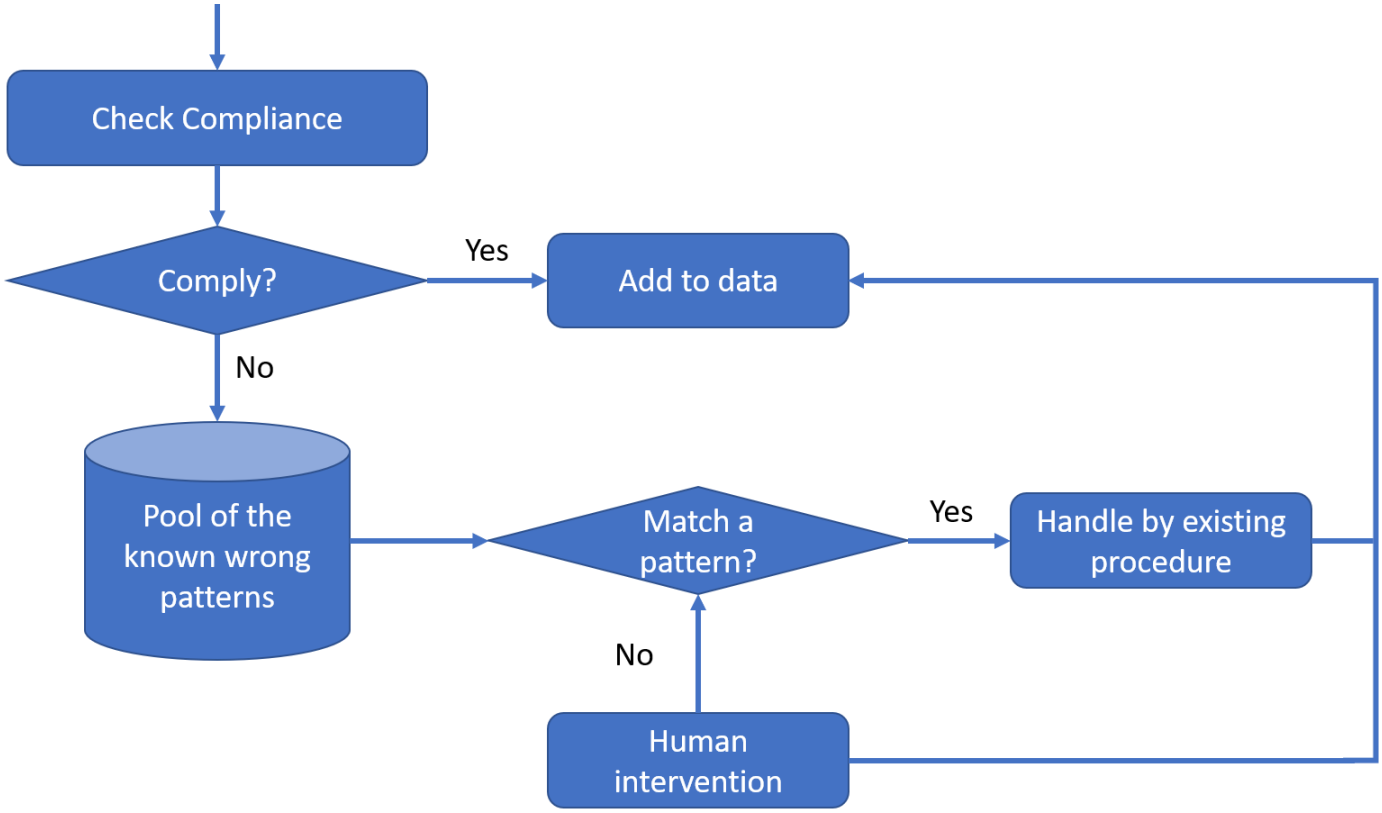
Error catching and warnings

- stop: throws an error, terminates programme
- warning: throws a warning, programme continues
- tryCatch: anything not caught by test cases, terminates without crashing programme
- finally: wrap up programme after error (e.g. closing connections)

```
GetMax0or1 = function(m, tie_breaker=1, by_row=T) {
  tryCatch ({
    # error catching
    allowed = c(0,1)
    ## m
    if (!is.matrix(m)) {
      stop("Illegal arguments: m is not a matrix\n")
    }
    if (sum(!(m %in% allowed)) > 0) {
      stop("Illegal arguments: m contains elements other than 0 and 1\n")
    }
    ## tie_breaker
    if (!(tie_breaker %in% allowed)) {
      stop("Illegal arguments: tie_breaker is neither 0 nor 1\n")
    }
    ## by_row
    if (!(by_row %in% allowed)) {
      stop("Illegal arguments: by_rows is not a logical input\n")
    }

    # warnings
    # matrix contains True/False -> convert to 1/0


    # check rows/col, depending on by_row
    sum = NA
    if (by_row) {
      sum = apply(m, 1, sum)
    } else {
      sum = apply(m, 2, sum)
    }

    # check majority, depending on tie breaker
    res = NA
    if (tie_breaker) {
      res = as.integer(sum >= ncol(m)/2)
    } else {
      res = as.integer(sum > ncol(m/2))
    }
    return(res)
  }, error=function(errorMessage){
    message(errorMessage)
  }, warning=function(warningMessage){
    message(warningMessage)
  }, finally={
  })
}

m = matrix(c(1,0,1,1,1,1,0,0,1,0,0,0), ncol=3, byrow=T)
GetMax0or1(m)
```

```
## [1] 1 1 0 0
```

Data Cleaning




```
families = read.csv("./Data/The family with the largest number of children.csv")

# match regex = "^([FM],)+[FM]$"
```

Tutorial 7

```
ks = read.csv("./Data/ks-projects.csv")
head(ks)
```

```
##           ID                                                 name
## 1 1000002330                      The Songs of Adelaide & Abullah
## 2 1000003930           Greeting From Earth: ZGAC Arts Capsule For ET
## 3 1000004038                                        Where is Hank?
## 4 1000007540       ToshiCapital Rekordz Needs Help to Complete Album
## 5 1000011046 Community Film Project: The Art of Neighborhood Filmmaking
## 6 1000014025                                  Monarch Espresso Bar
##         category main_category currency   deadline  goal            launched
## 1         Poetry    Publishing      GBP 2015-10-09  1000 2015-08-11 12:12:28
## 2 Narrative Film  Film & Video      USD 2017-11-01 30000 2017-09-02 04:43:57
## 3 Narrative Film  Film & Video      USD 2013-02-26 45000 2013-01-12 00:20:50
## 4          Music         Music      USD 2012-04-16  5000 2012-03-17 03:24:11
## 5   Film & Video  Film & Video      USD 2015-08-29 19500 2015-07-04 08:35:03
## 6    Restaurants          Food      USD 2016-04-01 50000 2016-02-26 13:38:27
##   pledged      state backers country usd.pledged usd_pledged_real usd_goal_real
## 1       0     failed       0      GB           0                0       1533.95
## 2    2421     failed      15      US         100             2421      30000.00
## 3     220     failed       3      US         220              220      45000.00
## 4       1     failed       1      US           1                1       5000.00
## 5    1283   canceled      14      US        1283             1283      19500.00
## 6   52375 successful     224      US       52375            52375      50000.00
```

```
# transform date type
ks$deadline = as.Date(ks$deadline)
ks$launched = as.Date(ks$launched)
ks$duration = ks$deadline - ks$launched
```

```
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
```

```
# only look at success and failures
ks_cleaned = ks[ks$state %in% c("successful", "failed"), ]
ks_cleaned$success = ks_cleaned$state == "successful"

# group by
df = ks_cleaned %>%
  group_by(main_category, country, duration) %>%
  summarise(success_rate=mean(success))
```

```
## `summarise()` has grouped output by 'main_category', 'country'. You can override using the
`.groups` argument.
```

```
head(df)
```

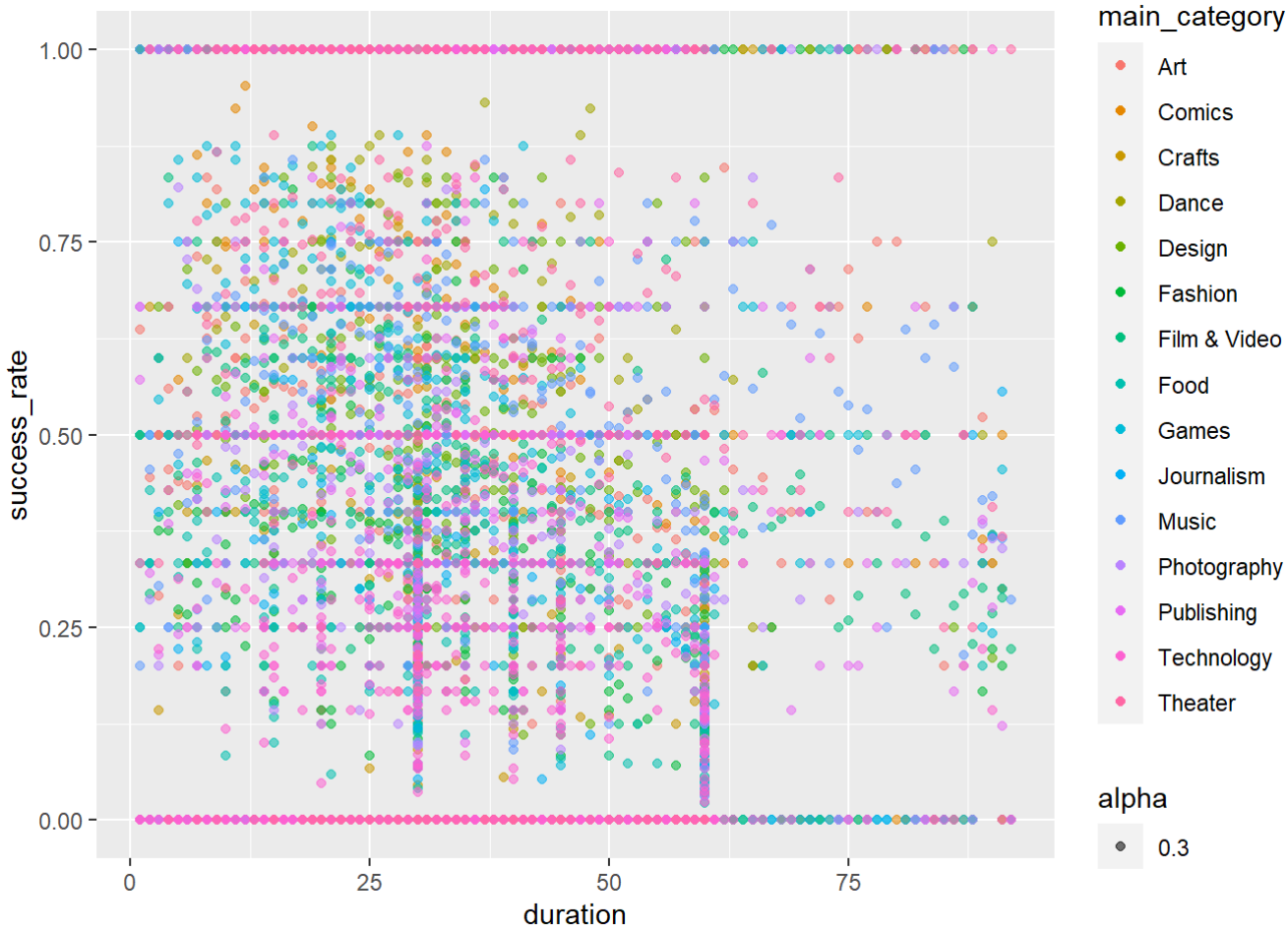
```
## # A tibble: 6 x 4
## # Groups:   main_category, country [1]
##   main_category country duration success_rate
##   <chr>         <chr>   <drtn>          <dbl>
## 1 Art           AT       3 days         1
## 2 Art           AT      13 days         0
## 3 Art           AT      21 days         0
## 4 Art           AT      28 days         0
## 5 Art           AT      30 days         0.118
## 6 Art           AT      31 days         0
```

```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
ggplot(data=df, aes(x=duration, y=success_rate, color=main_category, alpha=0.3)) + geom_point
()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to cont
inuous.
```


success_rate
1.00
0.75
0.50
0.25
0.00
0
25
50
75
duration
main_category
Art
Comics
Crafts
Dance
Design
Fashion
Film & Video
Food
Games
Journalism
Music
Photography
Publishing
Technology
Theater
alpha
0.3

Tutorial 8

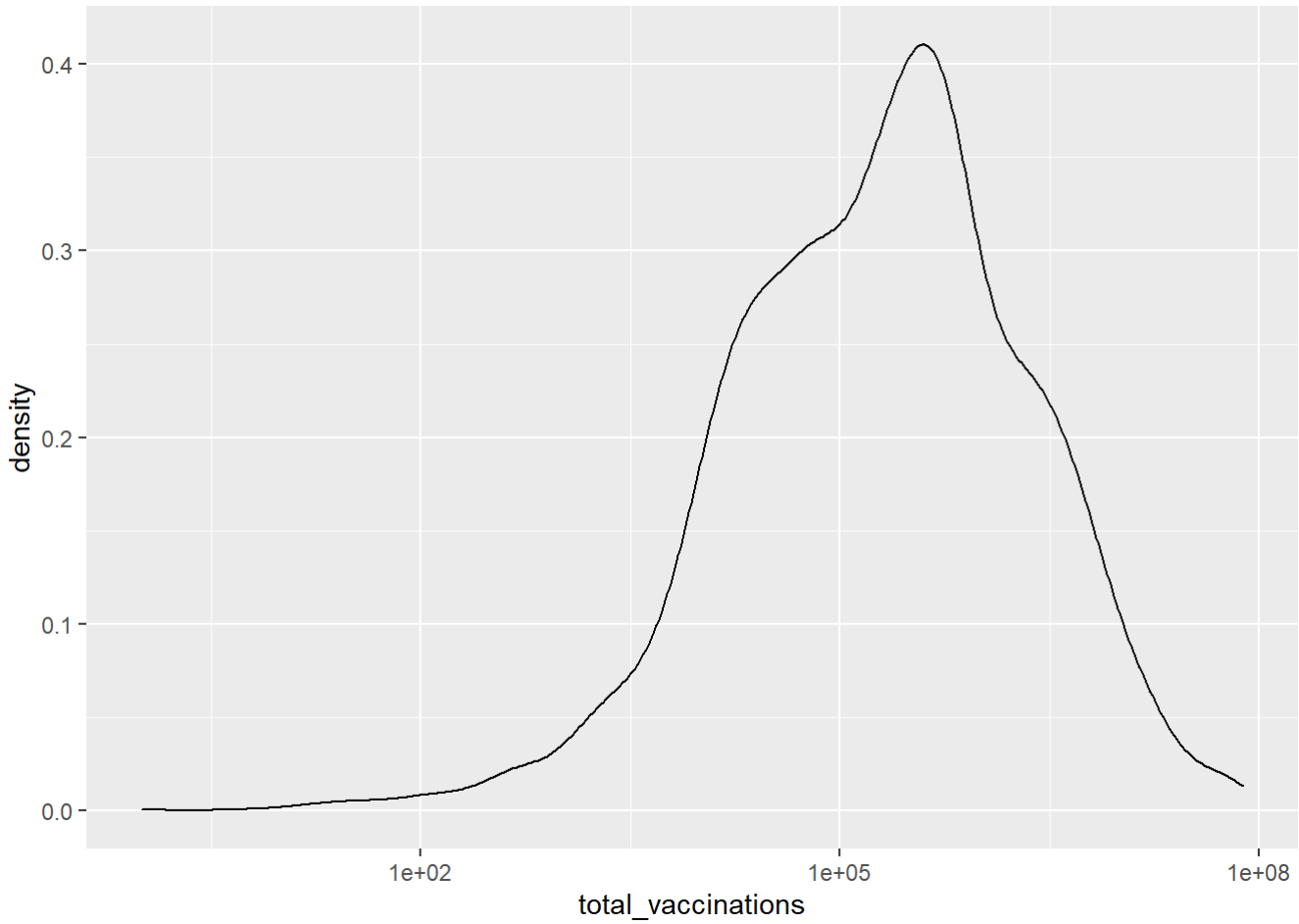
```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
data <- read.csv("./Data/country_vaccinations.csv")

data$date <- as.Date(data$date)
data$iso_code <- as.factor(data$iso_code)
g1 <- ggplot(data,aes(x=total_vaccinations))+geom_density()+scale_x_log10()
g1 # note that the axis ticks are difficult to interpret
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

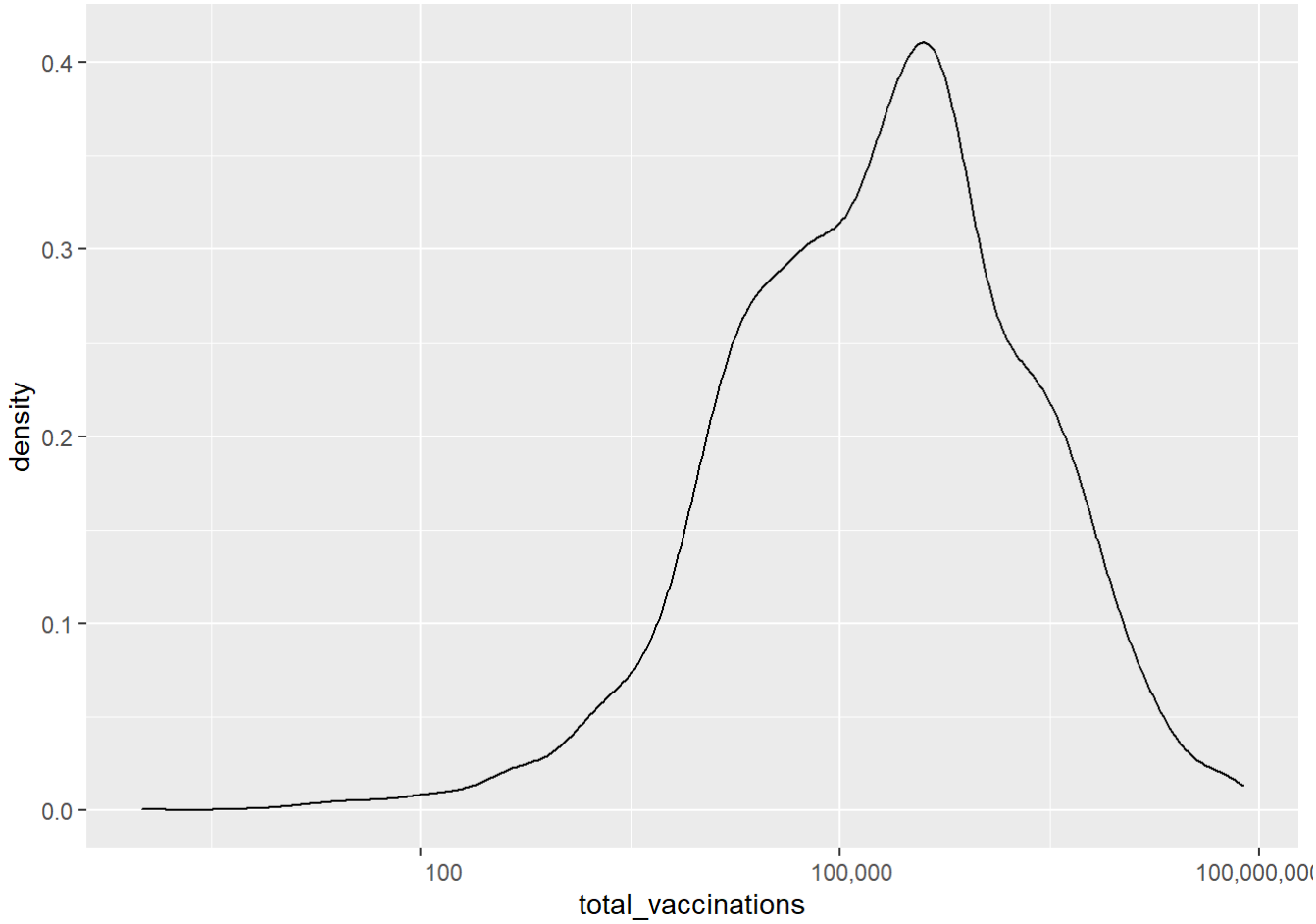
```
## Warning: Removed 1683 rows containing non-finite values (stat_density).
```



```
# transform data
formatX <- function(x)
{
  format(x, big.mark=",",scientific = F)
}

# labels will be formatted using the formatX function
g2 <- ggplot(data,aes(x=total_vaccinations))+geom_density()+scale_x_log10(labels=formatX)
g2 # labels are more interpretable BUT last number is truncated
```

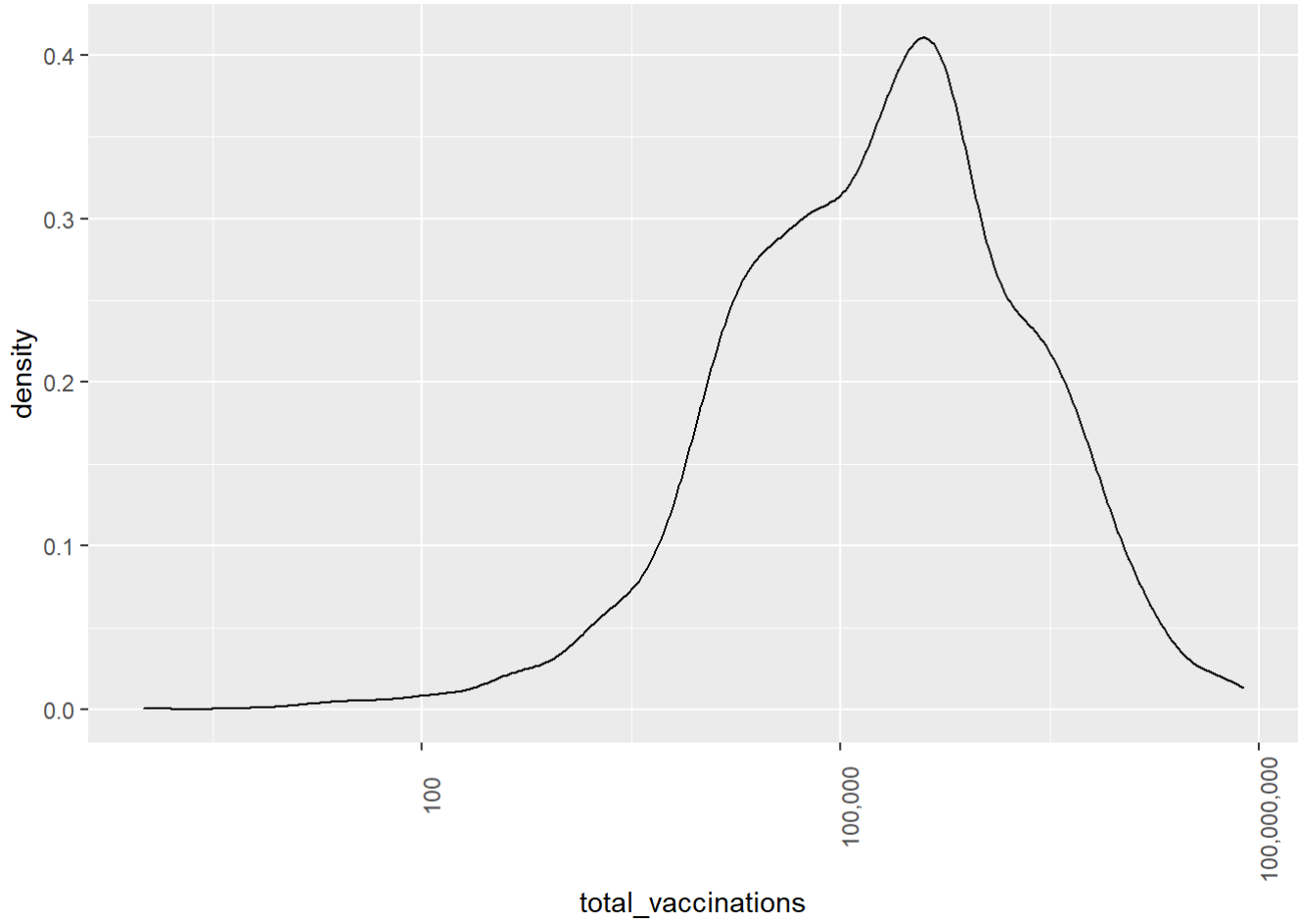
```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Removed 1683 rows containing non-finite values (stat_density).
```




```
g2 = g2 + theme(axis.text.x = element_text(angle = 90)) # roatate labels by 90 degrees
g2 # ticks have a lot of zeros
```

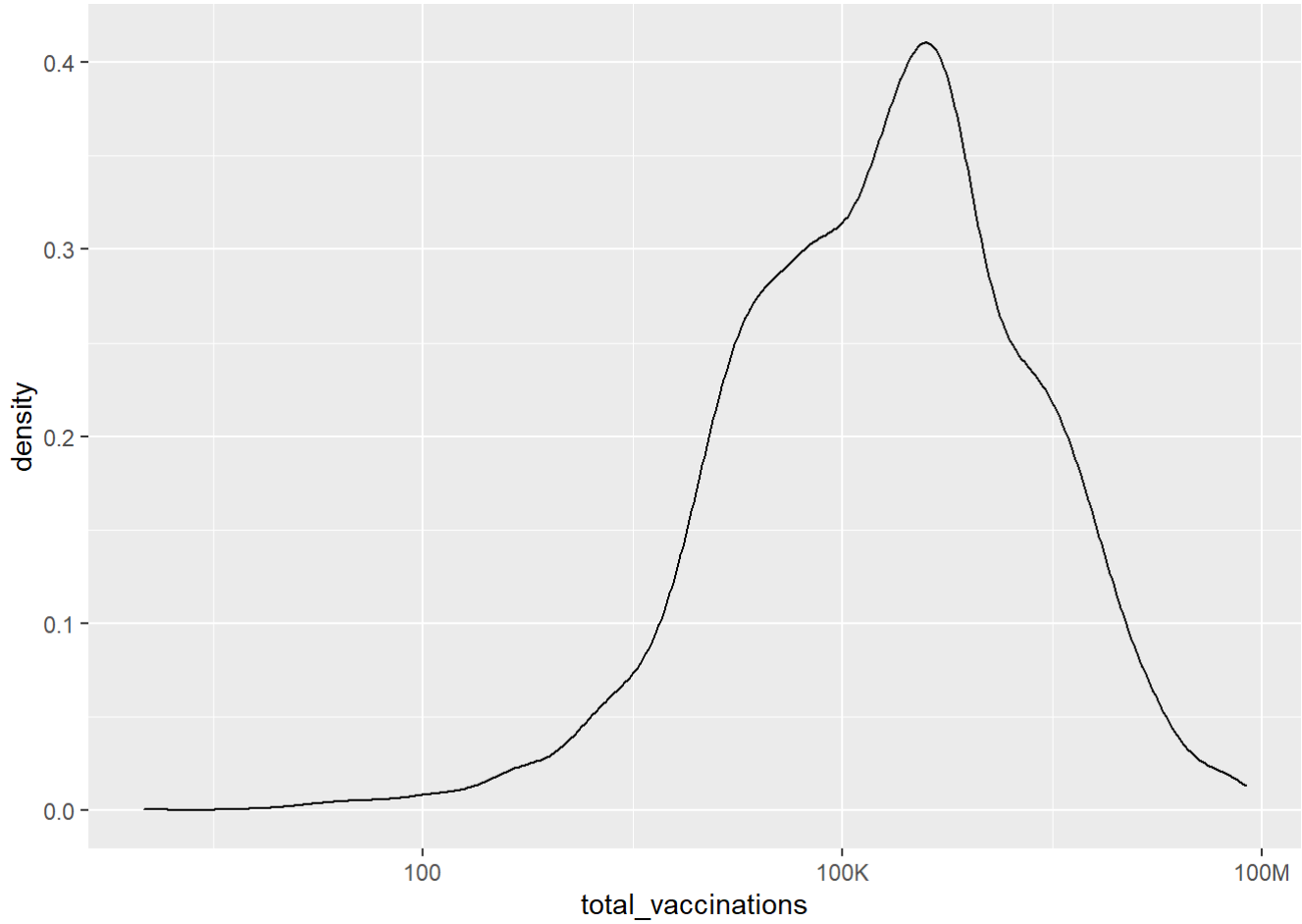
```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Removed 1683 rows containing non-finite values (stat_density).
```




```
# format function to adjust for the size
formatX0 <- function(x) {
  dplyr::case_when(
      x < 1e3 ~ as.character(x),
      x < 1e6 ~ paste0(as.character(x/1e3), "K"),
      x < 1e9 ~ paste0(as.character(x/1e6), "M"),
      x < 1e12 ~ paste0(as.character(x/1e9), "B"),
      TRUE ~ "To be implemented..."
  )
}

g3 <- ggplot(data,aes(x=total_vaccinations))+geom_density()+scale_x_log10(labels=formatX0)
g3
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Removed 1683 rows containing non-finite values (stat_density).
```




```
data = read.csv("./Data/flights1.csv")

# Format time
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.1
```

```
##
## Attaching package: 'lubridate'
```

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

```
data$date <- paste(data$YEAR,data$MONTH,data$DAY,sep = "-")

data$SCHEDULED_DEPARTURE1 <- substr(as.POSIXct(sprintf("%04.0f", data$SCHEDULED_DEPARTURE), f
ormat='%H%M'), 12, 19)
data$SCHEDULED_DEPARTURE_TIME <- strptime(paste(data$date,data$SCHEDULED_DEPARTURE1),format=
'%Y-%m-%d %H:%M:%S')
data$WHEELS_OFF_TIME <- strptime(paste(data$date,data$WHEELS_OFF1),format='%Y-%m-%d %H:%M:%S'
)

#
data$DEPARTURE_TIME1 <- substr(as.POSIXct(sprintf("%04.0f", data$DEPARTURE_TIME), format='%H%
M'), 12, 19)
data$ACTUAL_DEPARTURE_TIME <- strptime(paste(data$date,data$DEPARTURE_TIME1),format='%Y-%m-%d
%H:%M:%S')

data$WHEELS_OFF1 <- substr(as.POSIXct(sprintf("%04.0f", data$WHEELS_OFF), format='%H%M'), 12,
19)
data$WHEELS_OFF_TIME <- strptime(paste(data$date,data$WHEELS_OFF1),format='%Y-%m-%d %H:%M:%S'
)

#
data$WHEELS_ON1 <- substr(as.POSIXct(sprintf("%04.0f", data$WHEELS_ON), format='%H%M'), 12, 1
9)
data$WHEELS_ON_TIME <-  strptime(paste(data$date,data$WHEELS_ON1),format='%Y-%m-%d %H:%M:%S')

#
data$SCHEDULED_ARRIVAL1 <- substr(as.POSIXct(sprintf("%04.0f", data$SCHEDULED_ARRIVAL), forma
t='%H%M'), 12, 19)
data$SCHEDULED_ARRIVAL_TIME <- strptime(paste(data$date,data$SCHEDULED_ARRIVAL1),format='%Y-%
m-%d %H:%M:%S')

# data$ARRIVAL_TIME1 <- substr(as.POSIXct(sprintf("%04.0f", data$ARRIVAL_TIME), format='%H%
M'), 12, 19)
data$ACTUAL_ARRIVAL_TIME <- strptime(paste(data$date,data$ARRIVAL_TIME1),format='%Y-%m-%d %
H:%M:%S')

head(data)
```

```
##   YEAR MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER TAIL_NUMBER ORIGIN_AIRPORT
## 1 2015     1   1           4      F9           365      N218FR            MKE
## 2 2015     1   1           4      B6           746      N587JB            PSE
## 3 2015     1   1           4      F9          1338      N906FR            IAD
## 4 2015     1   1           4      OO          5536      N779SK            PSP
## 5 2015     1   1           4      B6          2324      N206JB            MCO
## 6 2015     1   1           4      DL          2499      N696DL            ATL
##   DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## 1                 DEN                 545            621              36
## 2                 JFK                 600            557              -3
## 3                 MSP                 620            609             -11
## 4                 IAH                 650            801              71
## 5                 DCA                 655            651              -4
## 6                 SLC                 720            719              -1
##   TAXI_OUT WHEELS_OFF SCHEDULED_TIME ELAPSED_TIME AIR_TIME DISTANCE WHEELS_ON
## 1        9        630            161          142      120      896       730
## 2        9        606            241          239      225     1617       851
## 3       12        621            165          161      142      908       743
## 4       10        811            178          162      147     1269      1238
## 5       14        705            124          120      103      759       848
## 6       31        750            260          255      218     1590       928
##   TAXI_IN SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY DIVERTED CANCELLED
## 1      13               726          743            17        0         0
## 2       5               901          856            -5        0         0
## 3       7               805          750           -15        0         0
## 4       5              1148         1243            55        0         0
## 5       3               859          851            -8        0         0
## 6       6               940          934            -6        0         0
##   CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY AIRLINE_DELAY
## 1                                   0              0            17
## 2                                  NA             NA            NA
## 3                                  NA             NA            NA
## 4                                   0              0             0
## 5                                  NA             NA            NA
## 6                                  NA             NA            NA
##   LATE_AIRCRAFT_DELAY WEATHER_DELAY     date SCHEDULED_DEPARTURE1
## 1                   0             0 2015-1-1             05:45:00
## 2                  NA            NA 2015-1-1             06:00:00
## 3                  NA            NA 2015-1-1             06:20:00
## 4                   0            55 2015-1-1             06:50:00
## 5                  NA            NA 2015-1-1             06:55:00
## 6                  NA            NA 2015-1-1             07:20:00
##   SCHEDULED_DEPARTURE_TIME     WHEELS_OFF_TIME DEPARTURE_TIME1
## 1      2015-01-01 05:45:00 2015-01-01 06:30:00        06:21:00
## 2      2015-01-01 06:00:00 2015-01-01 06:06:00        05:57:00
## 3      2015-01-01 06:20:00 2015-01-01 06:21:00        06:09:00
## 4      2015-01-01 06:50:00 2015-01-01 08:11:00        08:01:00
## 5      2015-01-01 06:55:00 2015-01-01 07:05:00        06:51:00
## 6      2015-01-01 07:20:00 2015-01-01 07:50:00        07:19:00
##   ACTUAL_DEPARTURE_TIME WHEELS_OFF1 WHEELS_ON1      WHEELS_ON_TIME
## 1   2015-01-01 06:21:00    06:30:00   07:30:00 2015-01-01 07:30:00
## 2   2015-01-01 05:57:00    06:06:00   08:51:00 2015-01-01 08:51:00
## 3   2015-01-01 06:09:00    06:21:00   07:43:00 2015-01-01 07:43:00
## 4   2015-01-01 08:01:00    08:11:00   12:38:00 2015-01-01 12:38:00
## 5   2015-01-01 06:51:00    07:05:00   08:48:00 2015-01-01 08:48:00
## 6   2015-01-01 07:19:00    07:50:00   09:28:00 2015-01-01 09:28:00
##   SCHEDULED_ARRIVAL1 SCHEDULED_ARRIVAL_TIME ACTUAL_ARRIVAL_TIME
```

```
## 1           07:26:00    2015-01-01 07:26:00             <NA>
## 2           09:01:00    2015-01-01 09:01:00             <NA>
## 3           08:05:00    2015-01-01 08:05:00             <NA>
## 4           11:48:00    2015-01-01 11:48:00             <NA>
## 5           08:59:00    2015-01-01 08:59:00             <NA>
## 6           09:40:00    2015-01-01 09:40:00             <NA>
```

Tutorial 9

Notes: - ggmap has geoplot function that can give the longitude and latitude from the city and street name

```
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("ggmap")
```

```
## Warning: package 'ggmap' was built under R version 4.1.1
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library("ggplot2")
library("tidyr")
library("lubridate")
```

```
## Warning: package 'lubridate' was built under R version 4.1.1
```

```
##
## Attaching package: 'lubridate'
```

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

```
data = read.csv("./Data/Police Use of Force.csv")
head(data)
```

```
##   INCIDENT_DATE INCIDENT_TIME    UOF_NUMBER OFFICER_ID OFFICER_GENDER
## 1      9/3/2016    4:14:00 AM         37702      10810           Male
## 2       3/22/16   11:00:00 PM         33413       7706           Male
## 3       5/22/16    1:29:00 PM         34567      11014           Male
## 4     1/10/2016    8:55:00 PM         31460       6692           Male
## 5     11/8/2016    2:30:00 AM  37879, 37898       9844           Male
## 6     9/11/2016    7:20:00 PM         36724       9855           Male
##   OFFICER_RACE OFFICER_HIRE_DATE OFFICER_YEARS_ON_FORCE OFFICER_INJURY
## 1        Black          5/7/2014                     2             No
## 2        White          1/8/1999                    17            Yes
## 3        Black           5/20/15                     1             No
## 4        Black           7/29/91                    24             No
## 5        White         10/4/2009                     7             No
## 6        White         6/10/2009                     7             No
##            OFFICER_INJURY_TYPE OFFICER_HOSPITALIZATION SUBJECT_ID SUBJECT_RACE
## 1 No injuries noted or visible                      No      46424        Black
## 2               Sprain/Strain                     Yes      44324     Hispanic
## 3 No injuries noted or visible                      No      45126     Hispanic
## 4 No injuries noted or visible                      No      43150     Hispanic
## 5 No injuries noted or visible                      No      47307        Black
## 6 No injuries noted or visible                      No      46549        White
##   SUBJECT_GENDER SUBJECT_INJURY          SUBJECT_INJURY_TYPE
## 1         Female            Yes      Non-Visible Injury/Pain
## 2           Male             No No injuries noted or visible
## 3           Male             No No injuries noted or visible
## 4           Male            Yes               Laceration/Cut
## 5           Male             No No injuries noted or visible
## 6         Female             No No injuries noted or visible
##   SUBJECT_WAS_ARRESTED SUBJECT_DESCRIPTION          SUBJECT_OFFENSE
## 1                  Yes   Mentally unstable                    APOWW
## 2                  Yes   Mentally unstable                    APOWW
## 3                  Yes             Unknown                    APOWW
## 4                  Yes FD-Unknown if Armed           Evading Arrest
## 5                  Yes             Unknown Other Misdemeanor Arrest
## 6                  Yes             Unknown                Assault/FV
##   REPORTING_AREA BEAT SECTOR      DIVISION LOCATION_DISTRICT STREET_NUMBER
## 1           2062  134    130       CENTRAL               D14           211
## 2           1197  237    230     NORTHEAST                D9          7647
## 3           4153  432    430     SOUTHWEST                D6           716
## 4           4523  641    640 NORTH CENTRAL               D11          5600
## 5           2167  346    340     SOUTHEAST                D7          4600
## 6           1134  235    230     NORTHEAST                D9          1234
##    STREET_NAME STREET_DIRECTION STREET_TYPE
## 1        Ervay                N         St.
## 2     Ferguson             NULL         Rd.
## 3 bimebella dr             NULL         Ln.
## 4          LBJ             NULL       Frwy.
## 5    Malcolm X                S       Blvd.
## 6        Peavy             NULL         Rd.
##   LOCATION_FULL_STREET_ADDRESS_OR_INTERSECTION LOCATION_CITY LOCATION_STATE
## 1                              211 N ERVAY ST        Dallas             TX
## 2                             7647 FERGUSON RD        Dallas             TX
## 3                             716 BIMEBELLA LN        Dallas             TX
## 4                               5600 L B J FWY        Dallas             TX
## 5                         4600 S MALCOLM X BLVD        Dallas             TX
## 6                                1234 PEAVY RD        Dallas             TX
##   LOCATION_LATITUDE LOCATION_LONGITUDE INCIDENT_REASON REASON_FOR_FORCE
```

```
## 1          32.78220          -96.79746          Arrest           Arrest
## 2          32.79898          -96.71749          Arrest           Arrest
## 3          32.73971          -96.92519          Arrest           Arrest
## 4                NA                 NA          Arrest           Arrest
## 5                NA                 NA          Arrest           Arrest
## 6          32.83753          -96.69557          Arrest           Arrest
##      TYPE_OF_FORCE_USED1 TYPE_OF_FORCE_USED2 TYPE_OF_FORCE_USED3
## 1  Hand/Arm/Elbow Strike
## 2            Joint Locks
## 3      Take Down - Group
## 4         K-9 Deployment
## 5         Verbal Command     Take Down - Arm
## 6 Hand Controlled Escort
##   TYPE_OF_FORCE_USED4 TYPE_OF_FORCE_USED5 TYPE_OF_FORCE_USED6
## 1
## 2
## 3
## 4
## 5
## 6
##   TYPE_OF_FORCE_USED7 TYPE_OF_FORCE_USED8 TYPE_OF_FORCE_USED9
## 1
## 2
## 3
## 4
## 5
## 6
##   TYPE_OF_FORCE_USED10 NUMBER_EC_CYCLES FORCE_EFFECTIVE
## 1                                 NULL             Yes
## 2                                 NULL             Yes
## 3                                 NULL             Yes
## 4                                 NULL             Yes
## 5                                 NULL         No, Yes
## 6                                 NULL             Yes
```

1. Convert incident date and time into date time format

```
# detect 4 digit year and 2 digit year
incident_dates = parse_date_time(data$INCIDENT_DATE, c("%-m/%-d/%Y", "%-m/%-d/%y"))
# detect time
incident_times = parse_date_time(data$INCIDENT_TIME, c("%-I:%M:%S %p","%I:%M:%S %p"))
```

```
## Warning: 10 failed to parse.
```

```
# combine
incident_dates = incident_dates %>% as_datetime()
incident_times = incident_times %>% as_datetime()
datestrings = incident_dates %>% strftime(format="%d/%m/%Y")

data$NEW_DATETIME = paste(datestrings,incident_times) %>% strptime(format="%d/%m/%Y %H:%M:%S
 %p", tz="GMT") %>% as.POSIXct()
```

2. Split force-effective column

```
num_cols = data$FORCE_EFFECTIVE %>%
  sapply(strsplit, ", ") %>%
  sapply(length) %>%
  max()
new_cols = paste("FORCE_EFFECTIVE", seq(1, num_cols), sep="_")
data = separate(data, FORCE_EFFECTIVE, into=new_cols, sep=", ")
```

```
## Warning: Expected 10 pieces. Missing pieces filled with `NA` in 2382 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

3. Fill up NA in longitude and latitude using address

```
ggmap::register_google(key='AIzaSyCnGFj3-tmhyhkx2Suxw3HNa6P0c0fjHc0')

missing = is.na(data$LOCATION_LATITUDE)

loc = paste(data$LOCATION_FULL_STREET_ADDRESS_OR_INTERSECTION, data$LOCATION_CITY, ", ")
loc = loc[missing]
loc = geocode(loc, output="latlon")
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=5600+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=18600+DALLAS+NORTH+TOLL
WAY+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=9500+POPPY+DR+Dallas+,&
key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=9500+POPPY+DR+Dallas+,&
key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=10100+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=10100+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=9400+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=6897+VALLEY+GLEN+DR+Dal
las+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2486+MAPLE+ROUTH+CONN+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=9200+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=9200+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4300+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=3315+PARROT+ST+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=900+WOOD+ST+Dallas+,&ke
y=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8051+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8051+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8102+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8102+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8102+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=8102+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4845+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=39690+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=39000+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=39690+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=39690+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=39690+L+B+J+FWY+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4848+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4848+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4848+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4846+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=7909+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=7909+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=5201+BARNES+BRIDGE+RD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2222+S+SAINT+AUGUSTINE+
RD+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=1003+CONDOR+DR+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=700+WOODALL+RODGERS+FWY
+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=3500+S+MALCOLM+X+BLVD+D
allas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2900+VICTORY+AVE+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2900+VICTORY+AVE+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2900+VICTORY+AVE+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2900+VICTORY+AVE+Dallas
+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=6950+MARVIN+D+LOVE+SERV
+E+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4600+ELSIE+FAYE+HEGGINS
+ST+Dallas+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=WOODALL+RODGERS+FWY+Dal
las+,&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=4122+L+B+J+FWY+Dallas+,
&key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=5100+SPUR+408+Dallas+,&
key=xxx-tmhyhkx2Suxw3HNa6P0c0fjHc0
```

```
data[missing, c("LOCATION_LATITUDE", "LOCATION_LONGITUDE")] = loc[c("lat", "lon")]
```

4. Convert number of EC cycles to consistent numbers

```
data$NUMBER_EC_CYCLES = gsub("NULL", NA, data$NUMBER_EC_CYCLES, fixed=T)
data$NUMBER_EC_CYCLES = data$NUMBER_EC_CYCLES %>%
  sapply(strsplit, ", ") %>%
  sapply(as.integer) %>%
  sapply(sum)

head(data)
```

```
##   INCIDENT_DATE INCIDENT_TIME    UOF_NUMBER OFFICER_ID OFFICER_GENDER
## 1      9/3/2016    4:14:00 AM         37702      10810           Male
## 2       3/22/16   11:00:00 PM         33413       7706           Male
## 3       5/22/16    1:29:00 PM         34567      11014           Male
## 4     1/10/2016    8:55:00 PM         31460       6692           Male
## 5     11/8/2016    2:30:00 AM  37879, 37898       9844           Male
## 6     9/11/2016    7:20:00 PM         36724       9855           Male
##   OFFICER_RACE OFFICER_HIRE_DATE OFFICER_YEARS_ON_FORCE OFFICER_INJURY
## 1        Black          5/7/2014                     2            No
## 2        White          1/8/1999                    17           Yes
## 3        Black           5/20/15                     1            No
## 4        Black           7/29/91                    24            No
## 5        White         10/4/2009                     7            No
## 6        White         6/10/2009                     7            No
##            OFFICER_INJURY_TYPE OFFICER_HOSPITALIZATION SUBJECT_ID SUBJECT_RACE
## 1 No injuries noted or visible                      No      46424        Black
## 2               Sprain/Strain                     Yes      44324     Hispanic
## 3 No injuries noted or visible                      No      45126     Hispanic
## 4 No injuries noted or visible                      No      43150     Hispanic
## 5 No injuries noted or visible                      No      47307        Black
## 6 No injuries noted or visible                      No      46549        White
##   SUBJECT_GENDER SUBJECT_INJURY          SUBJECT_INJURY_TYPE
## 1         Female            Yes      Non-Visible Injury/Pain
## 2           Male             No No injuries noted or visible
## 3           Male             No No injuries noted or visible
## 4           Male            Yes               Laceration/Cut
## 5           Male             No No injuries noted or visible
## 6         Female             No No injuries noted or visible
##   SUBJECT_WAS_ARRESTED SUBJECT_DESCRIPTION          SUBJECT_OFFENSE
## 1                  Yes   Mentally unstable                    APOWW
## 2                  Yes   Mentally unstable                    APOWW
## 3                  Yes             Unknown                    APOWW
## 4                  Yes FD-Unknown if Armed           Evading Arrest
## 5                  Yes             Unknown Other Misdemeanor Arrest
## 6                  Yes             Unknown                Assault/FV
##   REPORTING_AREA BEAT SECTOR      DIVISION LOCATION_DISTRICT STREET_NUMBER
## 1           2062  134    130       CENTRAL               D14           211
## 2           1197  237    230     NORTHEAST                D9          7647
## 3           4153  432    430     SOUTHWEST                D6           716
## 4           4523  641    640 NORTH CENTRAL               D11          5600
## 5           2167  346    340     SOUTHEAST                D7          4600
## 6           1134  235    230     NORTHEAST                D9          1234
##    STREET_NAME STREET_DIRECTION STREET_TYPE
## 1        Ervay                N         St.
## 2     Ferguson             NULL         Rd.
## 3 bimebella dr             NULL         Ln.
## 4          LBJ             NULL       Frwy.
## 5    Malcolm X                S       Blvd.
## 6        Peavy             NULL         Rd.
##   LOCATION_FULL_STREET_ADDRESS_OR_INTERSECTION LOCATION_CITY LOCATION_STATE
## 1                              211 N ERVAY ST        Dallas             TX
## 2                             7647 FERGUSON RD       Dallas             TX
## 3                             716 BIMEBELLA LN       Dallas             TX
## 4                               5600 L B J FWY       Dallas             TX
## 5                         4600 S MALCOLM X BLVD       Dallas             TX
## 6                               1234 PEAVY RD       Dallas             TX
##   LOCATION_LATITUDE LOCATION_LONGITUDE INCIDENT_REASON REASON_FOR_FORCE
```

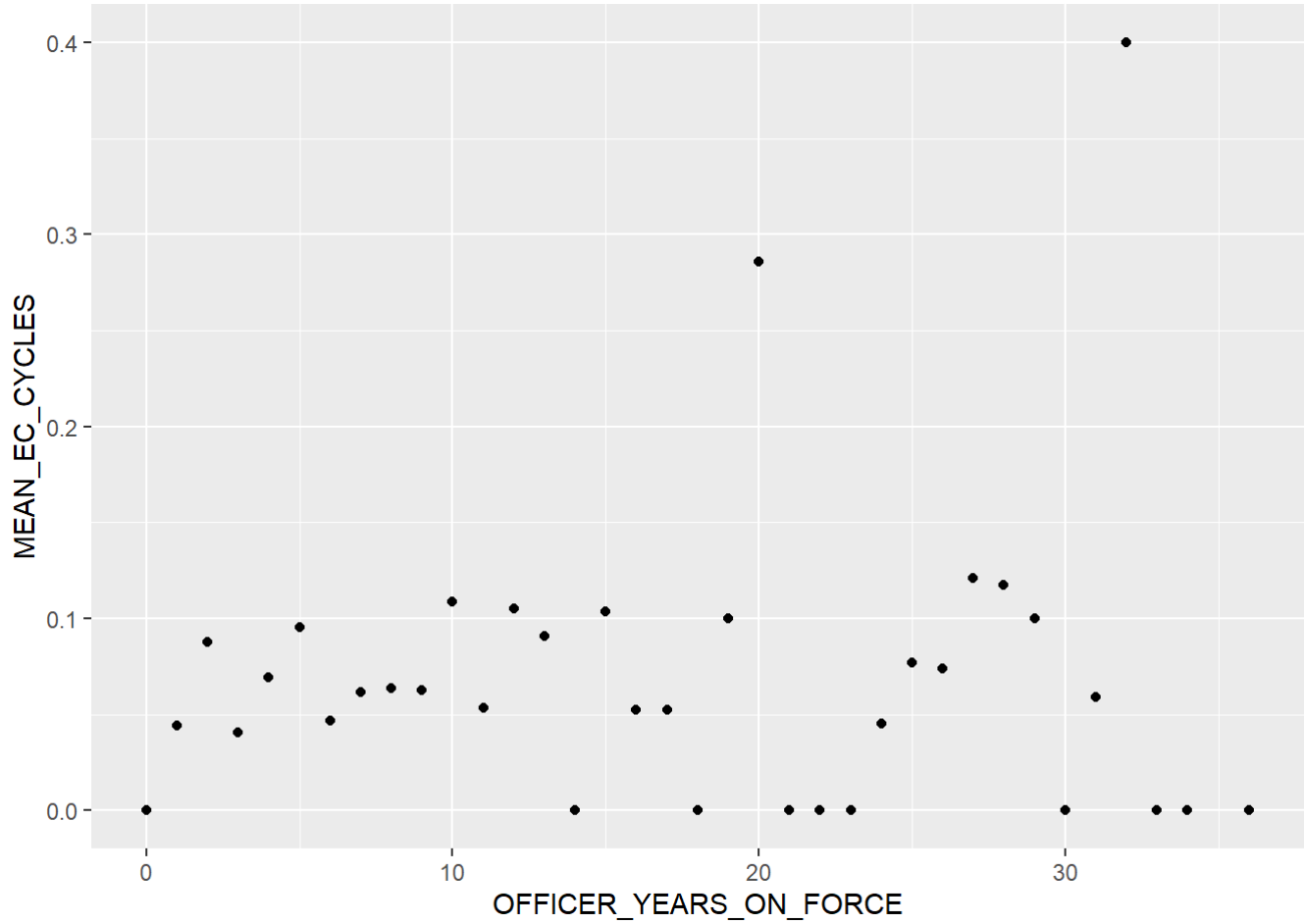
```
## 1          32.78220          -96.79746          Arrest           Arrest
## 2          32.79898          -96.71749          Arrest           Arrest
## 3          32.73971          -96.92519          Arrest           Arrest
## 4          32.92503          -96.80561          Arrest           Arrest
## 5          32.75657          -96.75320          Arrest           Arrest
## 6          32.83753          -96.69557          Arrest           Arrest
##      TYPE_OF_FORCE_USED1 TYPE_OF_FORCE_USED2 TYPE_OF_FORCE_USED3
## 1  Hand/Arm/Elbow Strike
## 2            Joint Locks
## 3      Take Down - Group
## 4         K-9 Deployment
## 5         Verbal Command     Take Down - Arm
## 6 Hand Controlled Escort
##   TYPE_OF_FORCE_USED4 TYPE_OF_FORCE_USED5 TYPE_OF_FORCE_USED6
## 1
## 2
## 3
## 4
## 5
## 6
##   TYPE_OF_FORCE_USED7 TYPE_OF_FORCE_USED8 TYPE_OF_FORCE_USED9
## 1
## 2
## 3
## 4
## 5
## 6
##   TYPE_OF_FORCE_USED10 NUMBER_EC_CYCLES FORCE_EFFECTIVE_1 FORCE_EFFECTIVE_2
## 1                                   NA               Yes              <NA>
## 2                                   NA               Yes              <NA>
## 3                                   NA               Yes              <NA>
## 4                                   NA               Yes              <NA>
## 5                                   NA                No               Yes
## 6                                   NA               Yes              <NA>
##   FORCE_EFFECTIVE_3 FORCE_EFFECTIVE_4 FORCE_EFFECTIVE_5 FORCE_EFFECTIVE_6
## 1              <NA>              <NA>              <NA>              <NA>
## 2              <NA>              <NA>              <NA>              <NA>
## 3              <NA>              <NA>              <NA>              <NA>
## 4              <NA>              <NA>              <NA>              <NA>
## 5              <NA>              <NA>              <NA>              <NA>
## 6              <NA>              <NA>              <NA>              <NA>
##   FORCE_EFFECTIVE_7 FORCE_EFFECTIVE_8 FORCE_EFFECTIVE_9 FORCE_EFFECTIVE_10
## 1              <NA>              <NA>              <NA>               <NA>
## 2              <NA>              <NA>              <NA>               <NA>
## 3              <NA>              <NA>              <NA>               <NA>
## 4              <NA>              <NA>              <NA>               <NA>
## 5              <NA>              <NA>              <NA>               <NA>
## 6              <NA>              <NA>              <NA>               <NA>
##   NEW_DATETIME
## 1         <NA>
## 2         <NA>
## 3         <NA>
## 4         <NA>
## 5         <NA>
## 6         <NA>
```

5. Data Visulisation

Racial Discrimination: - For the offense, compare the severity of the force used between different races - May also want to compare the officer race

When experience increases, is officer more likely to shoot:

```
data$OFFICER_YEARS_ON_FORCE = as.integer(data$OFFICER_YEARS_ON_FORCE)

obs = data %>% mutate_at(vars(NUMBER_EC_CYCLES), ~replace(., is.na(.), 0)) %>%
  group_by(OFFICER_YEARS_ON_FORCE) %>%
  summarise(MEAN_EC_CYCLES = mean(NUMBER_EC_CYCLES > 0))

ggplot(obs, aes(x=OFFICER_YEARS_ON_FORCE, y=MEAN_EC_CYCLES)) +
  geom_point()
```



Use ggmap/ leaflet to display the distribution of events: