problemset5

AUTHOR
Junyi Zhang

GitHub

https://github.com/juny1z/Problemset5.git

Problem 1

a.

```r
library(Rcpp)
setClass(
  "rational",
  slots = list(numerator = "numeric", denominator = "numeric"),
  prototype = list(numerator = 0, denominator = 1)
)

setMethod(
  "initialize", "rational",
  function(.Object, numerator, denominator) {
    if (denominator == 0) {
      stop("Denominator is non-zero.")
    }
    .Object@numerator <- numerator
    .Object@denominator <- denominator
    .Object
  }
)

setMethod(
  "show", "rational",
  function(object) {
    cat(object@numerator, "/", object@denominator, "\n")
  }
)

simplify <- function(object) {
  gcd <- function(a, b) {
    if (b == 0) return(a)
    gcd(b, a %% b)
  }
  g <- gcd(object@numerator, object@denominator)
  object@numerator <- object@numerator / g
  object@denominator <- object@denominator / g
  object
}

setGeneric("quotient", function(object, digits = 7) standardGeneric("quotient"))
```

```
[1] "quotient"
```

```
setMethod(
  "quotient", "rational",
  function(object, digits = 7) {
    result <- object@numerator / object@denominator
    round(result, digits)
  }
)

setMethod(
  "+", signature(e1 = "rational", e2 = "rational"),
  function(e1, e2) {
    num <- e1@numerator * e2@denominator + e2@numerator * e1@denominator
    den <- e1@denominator * e2@denominator
    simplify(new("rational", numerator = num, denominator = den))
  }
)

setMethod(
  "-", signature(e1 = "rational", e2 = "rational"),
  function(e1, e2) {
    num <- e1@numerator * e2@denominator - e2@numerator * e1@denominator
    den <- e1@denominator * e2@denominator
    simplify(new("rational", numerator = num, denominator = den))
  }
)

setMethod(
  "*", signature(e1 = "rational", e2 = "rational"),
  function(e1, e2) {
    num <- e1@numerator * e2@numerator
    den <- e1@denominator * e2@denominator
    simplify(new("rational", numerator = num, denominator = den))
  }
)

setMethod(
  "/", signature(e1 = "rational", e2 = "rational"),
  function(e1, e2) {
    num <- e1@numerator * e2@denominator
    den <- e1@denominator * e2@numerator
    simplify(new("rational", numerator = num, denominator = den))
  }
)
```

```
cppFunction('
  int gcdC(int a, int b) {
    if (b == 0) return abs(a);  // Ensure absolute value for negatives
    return gcdC(b, a % b);
  }

  int lcmC(int a, int b) {
    if (a == 0 || b == 0) return 0; // Handle edge case of zero
    return abs(a * b) / gcdC(a, b);
  }
')
```

b.

```
r1 <- new("rational", numerator = 24, denominator = 6)
r2 <- new("rational", numerator = 7, denominator = 230)
r3 <- new("rational", numerator = 0, denominator = 4)

r1 <- simplify(r1)
r2 <- simplify(r2)
r3 <- simplify(r3)

show(r1)
```

```
4 / 1
```

```
show(r2)
```

```
7 / 230
```

```
show(r3)
```

```
0 / 1
```

```
# This block intentionally produces an error
stop("This is an intentional error.")
```

```
Error: This is an intentional error.
```

```
r1
```

```
4 / 1
```

```
r3
```

```
0 / 1
```

```r
r1 + r2
```

```
927 / 230
```

```r
r1 - r2
```

```
913 / 230
```

```r
r1 * r2
```

```
14 / 115
```

```r
r1 / r2
```

```
920 / 7
```

```r
r1 + r3
```

```
4 / 1
```

```r
r1 * r3
```

```
0 / 1
```

```r
r2 / r3
```

```
Error in .local(.Object, ...): Denominator is non-zero.
```

```r
quotient(r1)
```

```
[1] 4
```

```r
quotient(r2)
```

```
[1] 0.0304348
```

```r
quotient(r2, digits = 3)
```

```
[1] 0.03
```

```r
quotient(r2, digits = 3.14)
```

```
[1] 0.03
```

```r
quotient(r2, digits = "avocado")
```

```
Error in round(result, digits): non-numeric argument to mathematical function
```

```r
q2 <- quotient(r2, digits = 3)
q2
```

```
[1] 0.03
```

```r
quotient(r3)
```

```
[1] 0
```

```r
simplify(r1)
```

```
4 / 1
```

```r
simplify(r2)
```

```
7 / 230
```

```r
simplify(r3)
```

```
0 / 1
```

c.

```r
library(methods)

setClass(
  "rational",
  slots = list(numerator = "numeric", denominator = "numeric"),
  prototype = list(numerator = 0, denominator = 1)
)

setMethod(
  "initialize", "rational",
  function(.Object, numerator, denominator) {
    if (denominator == 0) {
      stop("Error: Denominator is non-zero.")
    }
    if (!is.numeric(numerator) || !is.numeric(denominator)) {
      stop("Error: Both numerator and denominator must be numeric.")
    }
    .Object@numerator <- numerator
    .Object@denominator <- denominator
    .Object
  }
)

setMethod(
  "show", "rational",
  function(object) {
    cat(object@numerator, "/", object@denominator, "\n")
  }
)

simplify <- function(object) {
  gcd <- function(a, b) {
    if (b == 0) return(abs(a))
    gcd(b, a %% b)
  }
  g <- gcd(object@numerator, object@denominator)
  object@numerator <- object@numerator / g
  object@denominator <- object@denominator / g
  object
}

# Invalid: denominator is zero
tryCatch({
  r1 <- new("rational", numerator = 24, denominator = 0)
}, error = function(e) {
  print(e$message)
})
```

```
[1] "Error: Denominator is non-zero."
```

```r
# Invalid: numerator is string
tryCatch({
  r2 <- new("rational", numerator = "24", denominator = 6)
}, error = function(e) {
  print(e$message)
})
```

```
[1] "Error: Both numerator and denominator must be numeric."
```

```r
# Valid
tryCatch({
  r3 <- new("rational", numerator = 0, denominator = 4)
  show(r3)
}, error = function(e) {
  print(e$message)
})
```

```
0 / 4
```

Problem 2

a.

```
library(ggplot2)
art <- read.csv("/Users/zjyyy/Desktop/df_for_ml_improved_new_market.csv")
unique(art[, grep("^Genre", names(art))])
```

```
     Genre___Photography Genre___Print Genre___Sculpture Genre___Painting
1                      0             0                 0                1
2                      0             0                 1                0
5                      1             0                 0                0
123                    0             1                 0                0
1444                   0             0                 0                0
     Genre___Others
1                 1
2                 0
5                 0
123               0
1444              1
```

```
art$Genre___Others[art$Genre___Painting == 1] <- 0
unique(art[, grep("^Genre", names(art))])
```

```
     Genre___Photography Genre___Print Genre___Sculpture Genre___Painting
1                      0             0                 0                1
2                      0             0                 1                0
5                      1             0                 0                0
123                    0             1                 0                0
1444                   0             0                 0                0
     Genre___Others
1                 0
2                 0
5                 0
123               0
1444              1
```

```
art$genre <- "Photography"
art$genre[art$Genre___Print == 1] <- "Print"
art$genre[art$Genre___Sculpture == 1] <- "Sculpture"
art$genre[art$Genre___Painting == 1] <- "Painting"
art$genre[art$Genre___Others == 1] <- "Other"
table(art$genre)
```

```

      Other    Painting Photography       Print   Sculpture
         27         519        1746         414        1641
```

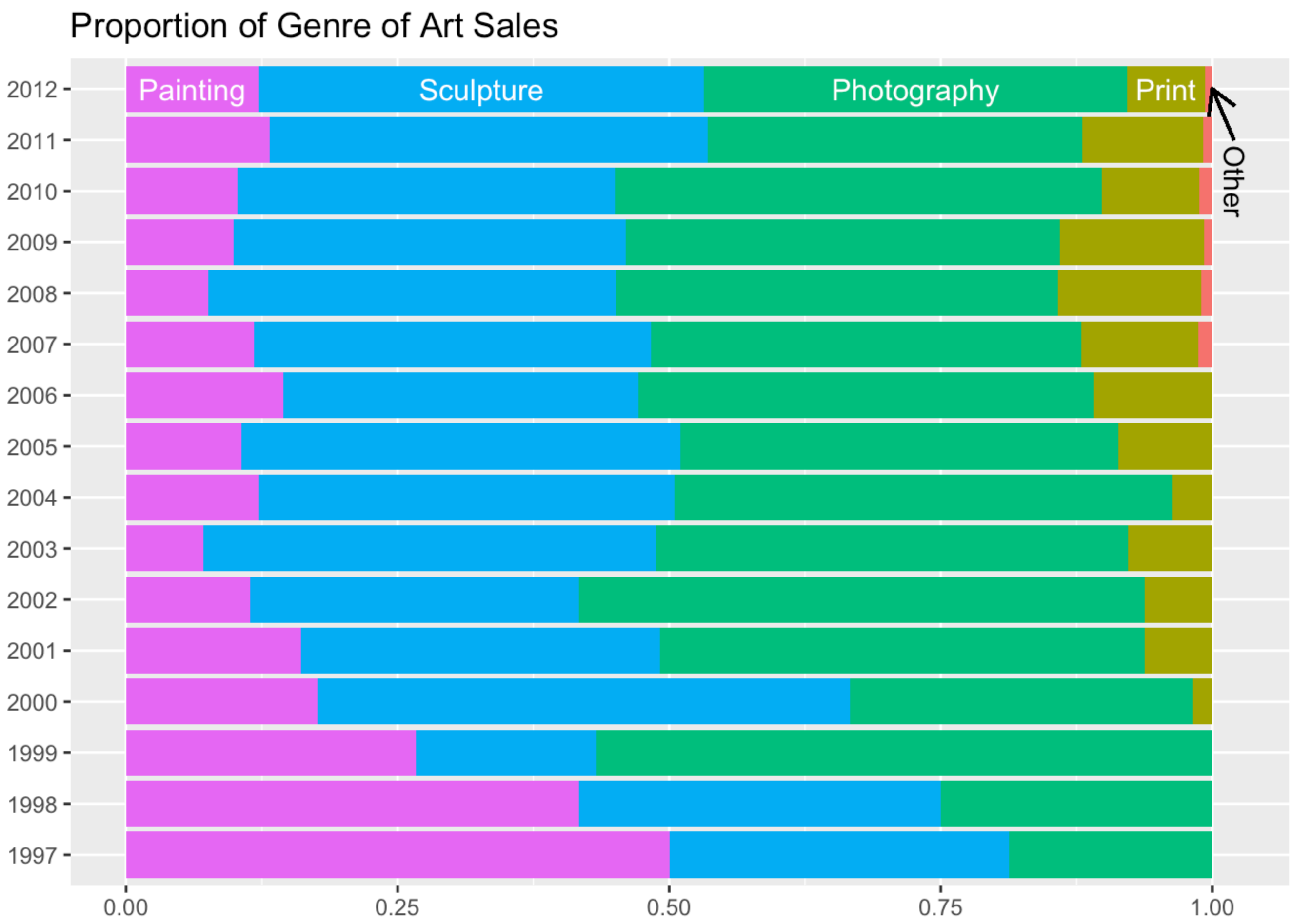
```
(yeargenre <- with(art, table(year, genre)))
```

```
      genre
year   Other Painting Photography Print Sculpture
  1997     0        8           3     0         5
  1998     0        5           3     0         4
  1999     0        8          17     0         5
  2000     0       19          34     2        53
  2001     0       18          50     7        37
  2002     0       11          50     6        29
  2003     0       12          73    13        70
  2004     0       23          86     7        72
  2005     0       32         122    26       122
  2006     0       57         165    43       129
  2007     5       47         158    43       146
  2008     4       31         166    54       153
  2009     3       41         165    55       149
  2010     5       42         184    37       143
  2011     6       95         247    80       289
  2012     4       70         223    41       235
```

```
ygperc <- yeargenre/apply(yeargenre, 1, sum)
ygperc <- ygperc[, c("Painting", "Sculpture", "Photography", "Print", "Other")]
ygpercm <- as.data.frame(ygperc)
# Reverse level of factors so ggplot draws it the same way
ygpercm$genre <- factor(ygpercm$genre, levels = rev(unique(ygpercm$genre)))
head(ygpercm)
```

```
  year    genre      Freq
1 1997 Painting 0.5000000
2 1998 Painting 0.4166667
3 1999 Painting 0.2666667
4 2000 Painting 0.1759259
5 2001 Painting 0.1607143
6 2002 Painting 0.1145833
```

```
ggplot(ygpercm, aes(y = Freq, x = year, fill = genre)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(y = NULL, x = NULL, title = "Proportion of Genre of Art Sales") +
  theme(legend.position = "off") +
  geom_text(data = ygpercm[ygpercm$year == 2012 & ygpercm$genre != "Other", ],
            aes(label = genre),
            position = position_stack(vjust = 0.5),
            color = "white",
            size = 4) +
  # Add the Other label
  geom_segment(aes(xend = 16, yend = 1, x = 15, y = 1.02),
               arrow = arrow(length = unit(0.15, "inches")),
             linewidth = .5, color = "black") +
  annotate("text", x = 14.9, y = 1.02, label = "Other", hjust = 0, angle = 270)
```



b.

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ lubridate 1.9.3     ✔ tibble    3.2.1
✔ purrr     1.0.2     ✔ tidyr     1.3.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```
library(plotly)
```

```

Attaching package: 'plotly'
```

```
The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

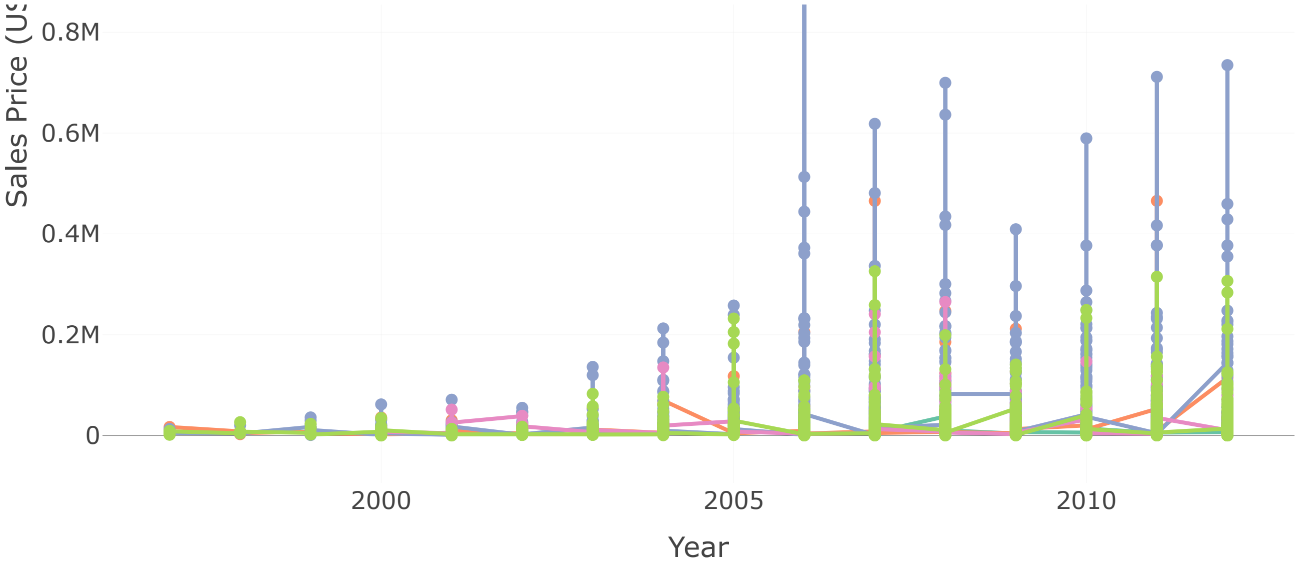
```r
genre_columns <- names(art)[grepl("Genre___", names(art))]

price_data <- art %>%
  select(year, price_usd, all_of(genre_columns)) %>%
  pivot_longer(cols = all_of(genre_columns),
               names_to = "Genre",
               values_to = "Count") %>%
  filter(Count == 1) %>%
  mutate(Genre = str_replace(Genre, "Genre___", ""))

interactive_plot <- price_data %>%
  plot_ly(
    x = ~year,
    y = ~price_usd,
    color = ~Genre,
    type = 'scatter',
    mode = 'markers+lines',
    hoverinfo = 'text',
    text = ~paste("Year:", year, "<br>Price (USD):", price_usd, "<br>Genre:", Genre)
  ) %>%
  layout(
    title = "Change in Sales Price Over Time by Genre",
    xaxis = list(title = "Year"),
    yaxis = list(title = "Sales Price (USD)"),
    legend = list(title = list(text = "Genre"))
  )

interactive_plot
```

Change in Sales Price Over Time by Genre
1.4M
1.2M
1M
SD)
Gen



Problem 3

a.

```
library(nycflights13)
library(data.table)
```

```

Attaching package: 'data.table'

The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:dplyr':

    between, first, last

The following object is masked from 'package:purrr':

    transpose
```

```
flights_dt <- as.data.table(flights)
airports_dt <- as.data.table(airports)

flights_dt <- merge(flights_dt, airports_dt[, .(faa, name)], by.x = "origin", by.y = "faa
setnames(flights_dt, "name", "origin_name")
flights_dt <- merge(flights_dt, airports_dt[, .(faa, name)], by.x = "dest", by.y = "faa",
setnames(flights_dt, "name", "dest_name")

# Departure Delay table
dep_delay_table <- flights_dt[, .(
  mean_dep_delay = mean(dep_delay, na.rm = TRUE),
  median_dep_delay = median(dep_delay, na.rm = TRUE),
  num_flights = .N
), by = origin_name][num_flights >= 10]

dep_delay_table <- dep_delay_table[order(-mean_dep_delay)]
print(dep_delay_table)
```

```
           origin_name mean_dep_delay median_dep_delay num_flights
                <char>          <num>            <num>       <int>
1: Newark Liberty Intl       15.10795               -1      120835
2: John F Kennedy Intl       12.11216               -1      111279
3:          La Guardia       10.34688               -3      104662
```

```
#Arrival Delay table
arr_delay_table <- flights_dt[, .(
  mean_arr_delay = mean(arr_delay, na.rm = TRUE),
  median_arr_delay = median(arr_delay, na.rm = TRUE),
  num_flights = .N
), by = dest_name][num_flights >= 10]

arr_delay_table <- arr_delay_table[order(-mean_arr_delay)]
print(arr_delay_table)
```

```
                              dest_name mean_arr_delay median_arr_delay
                                 <char>          <num>            <num>
 1:                 Columbia Metropolitan    41.76415094             28.0
 2:                            Tulsa Intl    33.65986395             14.0
 3:                     Will Rogers World    30.61904762             16.0
 4:                  Jackson Hole Airport    28.09523810             15.0
 5:                        Mc Ghee Tyson    24.06920415              2.0
 6:                 Dane Co Rgnl Truax Fld    20.19604317              1.0
 7:                         Richmond Intl    20.11125320              1.0
 8:          Akron Canton Regional Airport    19.69833729              3.0
 9:                       Des Moines Intl    19.00573614              0.0
10:                    Gerald R Ford Intl    18.18956044              1.0
11:                       Birmingham Intl    16.87732342             -2.0
12:         Theodore Francis Green State    16.23463687              1.0
13: Greenville-Spartanburg International    15.93544304             -0.5
14:    Cincinnati Northern Kentucky Intl    15.36456376             -3.0
15:             Savannah Hilton Head Intl    15.12950601             -1.0
16:           Manchester Regional Airport    14.78755365             -3.0
17:                          Eppley Afld    14.69889841             -2.0
18:                               Yeager    14.67164179             -1.5
19:                   Kansas City Intl    14.51405836              0.0
20:                         Albany Intl    14.39712919             -4.0
21:               General Mitchell Intl    14.16722038              0.0
22:                     Piedmont Triad    14.11260054             -2.0
23:              Washington Dulles Intl    13.86420212             -3.0
24:               Cherry Capital Airport    12.96842105            -10.0
```

```
25:              James M Cox Dayton Intl    12.68048606             -3.0
26:     Louisville International Airport    12.66938406             -2.0
27:                  Chicago Midway Intl    12.36422360             -1.0
28:                      Sacramento Intl    12.10992908              4.0
29:                    Jacksonville Intl    11.84483416             -2.0
30:                       Nashville Intl    11.81245891             -2.0
31:                Portland Intl Jetport    11.66040210             -4.0
32:               Greater Rochester Intl    11.56064461             -5.0
33:      Hartsfield Jackson Atlanta Intl    11.30011285             -1.0
34:                Lambert St Louis Intl    11.07846451             -3.0
35:                         Norfolk Intl    10.94909344             -4.0
36:            Baltimore Washington Intl    10.72673385             -5.0
37:                         Memphis Intl    10.64531435             -2.5
38:                   Port Columbus Intl    10.60132291             -3.0
39:                  Charleston Afb Intl    10.59296847             -4.0
40:                    Philadelphia Intl    10.12719014             -3.0
41:                  Raleigh Durham Intl    10.05238095             -3.0
42:                    Indianapolis Intl     9.94043412             -3.0
43:            Charlottesville-Albemarle     9.50000000             -5.0
44:               Cleveland Hopkins Intl     9.18161129             -5.0
45:        Ronald Reagan Washington Natl     9.06695204             -2.0
46:                      Burlington Intl     8.95099602             -4.0
47:                 Buffalo Niagara Intl     8.94595186             -5.0
48:                Syracuse Hancock Intl     8.90392501             -5.0
49:                          Denver Intl     8.60650021             -2.0
50:                      Palm Beach Intl     8.56297210             -3.0
51:                             Bob Hope     8.17567568             -3.0
52:       Fort Lauderdale Hollywood Intl     8.08212154             -3.0
53:                          Bangor Intl     8.02793296             -9.0
54:           Asheville Regional Airport     8.00383142             -1.0
55:                      Pittsburgh Intl     7.68099053             -5.0
56:                       Gallatin Field     7.60000000             -2.0
57:                 NW Arkansas Regional     7.46572581             -2.0
58:                           Tampa Intl     7.40852503             -4.0
59:               Charlotte Douglas Intl     7.36031885             -3.0
60:             Minneapolis St Paul Intl     7.27016886             -5.0
61:                      William P Hobby     7.17618819             -4.0
62:                         Bradley Intl     7.04854369            -10.0
63:                     San Antonio Intl     6.94537178             -9.0
64:                      South Bend Rgnl     6.50000000             -3.5
65:     Louis Armstrong New Orleans Intl     6.49017497             -6.0
66:                        Key West Intl     6.35294118              7.0
67:                        Eagle Co Rgnl     6.30434783             -4.0
68:                Austin Bergstrom Intl     6.01990875             -5.0
69:                   Chicago Ohare Intl     5.87661475             -8.0
70:                         Orlando Intl     5.45464309             -5.0
71:               Detroit Metro Wayne Co     5.42996346             -7.0
72:                        Portland Intl     5.14157973             -5.0
73:                        Nantucket Mem     4.85227273             -3.0
74:                      Wilmington Intl     4.63551402             -7.0
75:                    Myrtle Beach Intl     4.60344828            -13.0
```

```
  76:    Albuquerque International Sunport     4.38188976             -5.5
  77:         George Bush Intercontinental     4.24079040             -5.0
  78:        Norman Y Mineta San Jose Intl     3.44817073             -7.0
  79:               Southwest Florida Intl     3.23814963             -5.0
  80:                       San Diego Intl     3.13916574             -5.0
  81:              Sarasota Bradenton Intl     3.08243131             -5.0
  82:            Metropolitan Oakland Intl     3.07766990             -9.0
  83:                                 <NA>     3.01233913             -5.0
  84:   General Edward Lawrence Logan Intl     2.91439222             -9.0
  85:                   San Francisco Intl     2.67289152             -8.0
  86:                         Yampa Valley     2.14285714              2.0
  87:              Phoenix Sky Harbor Intl     2.09704733             -6.0
  88:            Montrose Regional Airport     1.78571429            -10.5
  89:                     Los Angeles Intl     0.54711094             -7.0
  90:               Dallas Fort Worth Intl     0.32212685             -9.0
  91:                           Miami Intl     0.29905978             -9.0
  92:                       Mc Carran Intl     0.25772849             -8.0
  93:                  Salt Lake City Intl     0.17625459             -8.0
  94:                           Long Beach    -0.06202723            -10.0
  95:                Martha\\\\'s Vineyard    -0.28571429            -11.0
  96:                  Seattle Tacoma Intl    -1.09909910            -11.0
  97:                        Honolulu Intl    -1.36519258             -7.0
  98:            John Wayne Arpt Orange Co    -7.86822660            -11.0
  99:                   Palm Springs Intl   -12.72222222            -13.5
                                 dest_name mean_arr_delay median_arr_delay
      num_flights
            <int>
   1:         116
   2:         315
   3:         346
   4:          25
   5:         631
   6:         572
   7:        2454
   8:         864
   9:         569
  10:         765
  11:         297
  12:         376
  13:         849
  14:        3941
  15:         804
  16:        1009
  17:         849
  18:         138
  19:        2008
  20:         439
  21:        2802
  22:        1606
  23:        5700
  24:         101
```

```
    25:        1525
    26:        1157
    27:        4113
    28:         284
    29:        2720
    30:        6333
    31:        2352
    32:        2416
    33:       17215
    34:        4339
    35:        1536
    36:        1781
    37:        1789
    38:        3524
    39:        2884
    40:        1632
    41:        8163
    42:        2077
    43:          52
    44:        4573
    45:        9705
    46:        2589
    47:        4681
    48:        1761
    49:        7266
    50:        6554
    51:         371
    52:       12055
    53:         375
    54:         275
    55:        2875
    56:          36
    57:        1036
    58:        7466
    59:       14064
    60:        7185
    61:        2115
    62:         443
    63:         686
    64:          10
    65:        3799
    66:          17
    67:         213
    68:        2439
    69:       17283
    70:       14082
    71:        9384
    72:        1354
    73:         265
    74:         110
    75:          59
```

```
76:         254
77:        7198
78:         329
79:        3537
80:        2737
81:        1211
82:         312
83:        7602
84:       15508
85:       13331
86:          15
87:        4656
88:          15
89:       16174
90:        8738
91:       11728
92:        5997
93:        2467
94:         668
95:         221
96:        3923
97:         707
98:         825
99:          19
    num_flights
```

b.

```r
flights_dt <- as.data.table(flights)
planes_dt <- as.data.table(planes)

flights_dt <- merge(flights_dt, planes_dt, by = "tailnum", all.x = TRUE)

fastest_aircraft <- flights_dt[
  !is.na(air_time) & air_time > 0 & !is.na(distance),
  .(
    avgmph = mean(distance / (air_time / 60), na.rm = TRUE),
    nflights = .N
  ),
  by = model
][order(-avgmph)][1]

print(fastest_aircraft)
```

```
     model   avgmph nflights
    <char>    <num>    <int>
1: 777-222 482.6254        4
```