Cheery Blossom 10-mile running race Analysis

Zhewen Shi

April 12, 2015

The data can be downloaded at http://cherryblossom.org/. I only analyse data between 1999 and 2010.

1. Read the data into R

There are 24 files which contain men and women data between 1999 and 2010. We should read the data into R and put them in one data frame for further analysis. The steps are as follows:

Algorithm 1 Main Function

1: *dirs* ← names of all data files
2: **for** i in range 1:length of dirs **do**
3: *dat*[[*i*]] ← readFile(dirs[i]), put the data of each file in one data frame
4: **end for**
5: *mileDat* ← merge all dat[[i]] together
6: saving mileDat in 'data.rda'

There is an import function in the third line above (**readFile(filename)**). It can be used to put the data of each file in one data frame. The data files are in a slightly non-standard format. Although we can easily read them by lines, we should analyze the structure of each line for further analysis. For this purpose, I choose to use the line full of = and blanks to split data in each line into different columns. Names of columns can be extracted using the previous line of line contains = and blanks.

Algorithm 2 readFile Function

Require: File name
Ensure: res: a data frame which contains data information in the file
1: **function** READFILE(*filename*)
2: *dataFM* ← readLines(filename), read all lines and put the data into dataFM
3: *dataFM* ← preProcess(dataFM, filename)
4: *res* ← breakLines(dataFM)
5: Add two columns in res, one stands for gender and the other stands for year
6: return (res)
7: **end function**

Algorithm 2 describes the **readFile** function, in which the two important subfunction is in line 2 and line 3. The function **preProcess(dataFM, filename)** is used to delete useless information and put dataFM(lines in the file) into same format, the function **breakLines(dataFM)** is used to read the data into a data frame.

1.1 R function preProcess(s, filename)

This function has two inputs, s and filename. **s** is a list in which each element stands for one line in the file. The output is modified **s**. The useless data are deleted. s[1] is the line contains column names. s[2] contains = and blanks to separate different cloumns. Other lines contain data.

The procedure is in Algorithm 3. The steps 1 to 7 is to locate the row number of the line only with = and blanks. If there is no such a line, use the file in the same year to get the line and add the header to s. The steps 8 to 13 are used to transfer header information into same format. Other steps are for deleting useless information.

In step 12, we assume that in previous line of offset line, If there is a blank and the next digit is a character, the position potentially stands for the end of one column. If there are only blanks in this position in other lines, it actually stands for the end of one column and in the offset line, the same position should be blank. So, if in the offset line, there is = in the same position, it shows that two columns are not separated by blank in the offset line and I will change the = to blank.

Algorithm 3 preProcess(s, filename)

1: offset← the row number of the line with = and blanks in s
2: **if** no offset can be found **then**
3: Transfer the file name to other file names in the same year
4: offset ← the row number of the line with = and blanks in the file of the same year
5: Copy the offset line and the previous line of the same year file to s
6: offset ← the row number of the line with = and blanks in s
7: **end if**
8: s[offset] ← change '/' by one blank if there are '/'s
9: s[offset - 1] ← change '/' by one blank if there are '/'s
10: s[offset] ← change the first blank to = if there are two continuous blanks
11: s[offset - 1] ← change the first blank behind digits such as 5 and 10 to -
12: s[offset] ← change the = to blank if the next = stands for the start of a new column
13: s[offset] ← change the first blank to = if there are two continuous blanks
14: **if** offset larger than 2 **then**
15: $s \leftarrow$ s[-(1:(offset - 2))]
16: **end if**
17: $n \leftarrow$ length(s)
18: **while** nchar(s[n]) == 0 or the data before the first blank of s[n] is not integer number **do**
19: $n \leftarrow n - 1$
20: **end while**
21: $s \leftarrow s[1 : n]$
22: $s \leftarrow$ s[which(deleteBlank(s) != '')], delete blank lines in s
23: return(s)

1.2 R function breakLines(s)

The input of this function is the output of previous function preProcess(s, filename), the output is a data frame which contains all useful data in the file. The procedure is in Algorithm 4. The small sub-functions such as how to change time format to seconds, how to use the same column name(use switch function in R) and how to delete blanks and tabs are in appendix 1.

Algorithm 4 breakLines(s)

fieldsList← a list begins with 0, i-th elements stands for the end of (i-1)-th column
colnm ← column names extracted from s[1] which is splited by the index in fieldsList
m ← split all rows in s[3:] by the index of fieldsList
res ← change m to data.frame, each row stands for a row in s[3:], the column name is column
res$*timeoguide* ← a column stands for under USATF open guideline or Age-Group guideline
for i in range 1: (number of columns -1) **do**
    if res[, i] contains all integer numbers **then**
        $res[, i] \leftarrow$ as.numeric(res[,i])
    else if res[, i] contains ':' **then**
        $res[, i] \leftarrow$ change the time format to seconds
        $res[,'timeoguide'] \leftarrow$ from the last digit of this column to input under which guideline

Algorithm 4 breakLines(s) (continued)

    else
        $res[, i] \leftarrow$ as.character(res[,i])
    end if
end for
return(res)

2. Data Analysis

After loading data into R, we can get a data frame which has 113190 rows and 16 columns. The summary of this data frame is as follows:

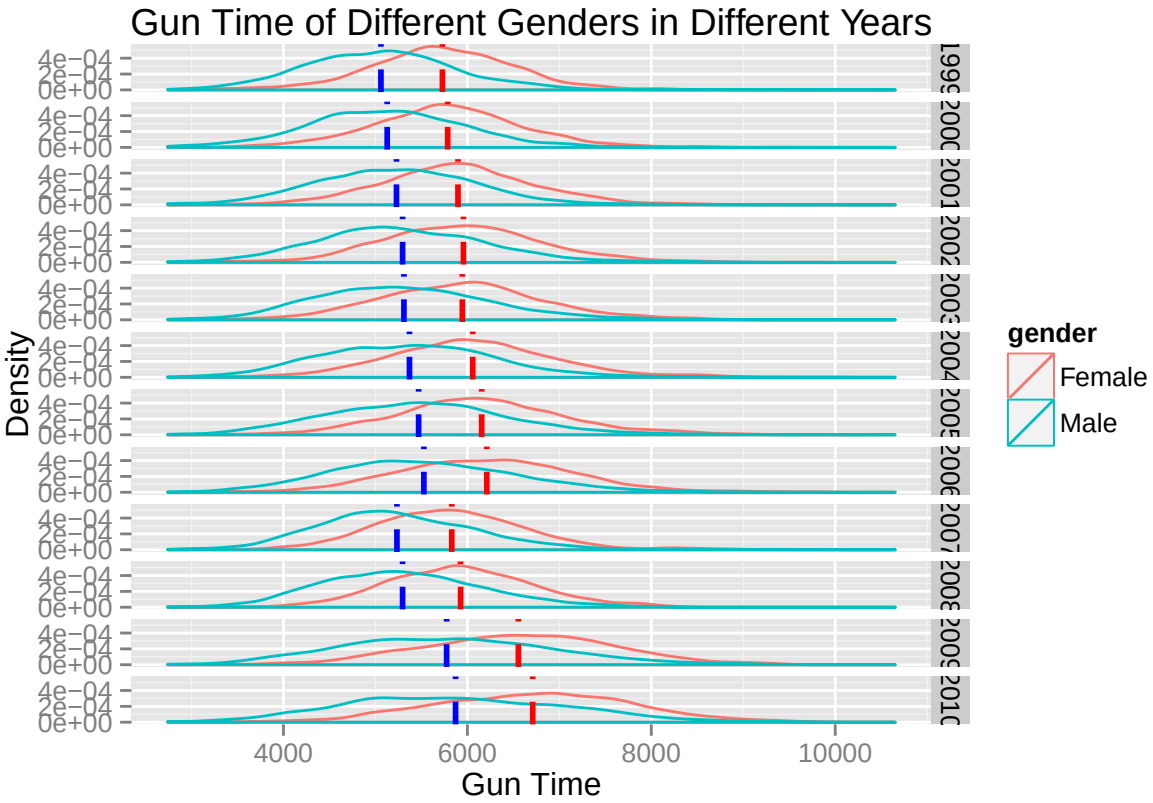
```
##      place         division          total           name
##  Min.   :   1   Min.   :   1.0   Min.   :   1    Length:113190
##  1st Qu.:1180   1st Qu.: 181.0   1st Qu.: 561    Class :character
##  Median :2367   Median : 462.0   Median :1150    Mode  :character
##  Mean   :2680   Mean   : 717.9   Mean   :1435
##  3rd Qu.:3803   3rd Qu.:1020.0   3rd Qu.:2242
##  Max.   :8853   Max.   :4069.0   Max.   :4069
##                 NA's   :13633    NA's   :13633
##       age          hometown             guntim           pace
##  Min.   : 0.00   Length:113190      Min.   : 2743   Min.   : 275.0
##  1st Qu.:28.00   Class :character   1st Qu.: 5058   1st Qu.: 502.0
##  Median :34.00   Mode  :character   Median : 5758   Median : 565.0
##  Mean   :36.33                      Mean   : 5806   Mean   : 567.4
##  3rd Qu.:43.00                      3rd Qu.: 6490   3rd Qu.: 629.0
##  Max.   :87.00                      Max.   :10651   Max.   :1061.0
##  NA's   :41                         NA's   :112     NA's   :34425
##  timeoguide        gender           year            num
##  FALSE:   800   Female:57302   2010   :15762   Min.   :    0
##  TRUE :   306   Male  :55888   2009   :14972   1st Qu.: 3764
##  NA's :112084                  2008   :12302   Median : 7768
##                                2007   :10964   Mean   : 8327
##                                2006   :10670   3rd Qu.:11924
##                                2005   : 8657   Max.   :57888
##                                (Other):39863   NA's   :14206
##      nettim           s                 5-mi           10-km
```

```
##  Min.    :   90   Length:113190      Min.    : 777    Min.    :1725
##  1st Qu.: 4917   Class :character   1st Qu.:2631    1st Qu.:3140
##  Median : 5514   Mode  :character   Median :2984    Median :3525
##  Mean    : 5549                      Mean    :3011    Mean    :3508
##  3rd Qu.: 6124                      3rd Qu.:3334    3rd Qu.:3887
##  Max.    :10651                     Max.    :6832    Max.    :5540
##  NA's    :28814                     NA's    :74858   NA's    :101413
```

'place' means people's rank in different years. 'division' means rank of people in his or her group in different years. 'total' means total number of athletes in his or her group. 'name' is the athlete's name. 'age' is his or her age. 'hometown' is his or her country. 'guntim' stands for gun time, the times used to finish the run. 'pace' is the time used when finishing a part of the whole run. There are three values in 'timeoguide', TRUE means under USATF OPEN guideline, FALSE means under USATF Age-Group guideline, NA means unknown. 'nettim' is another kind of timing pattern. '5-mi' and '10-km' are time used for a part of the whole run.

2.1 Gun time distribution of different genders in different years.

The following is the density plot of gun time in different years, in which gender 0 means female and 1 stands for male. We can see that men have less gun time than women. However, the mean gun time increases over years(except 2007 and 2008 it is a little better) both men and women. From the appendix 2, we can see that the age of participants in different years have the samilar distribution. Maybe people are taking less and less physical exercise these years and most people's physical condition are becoming worse. Their is another possibility that people just take this competition for fun. They do not care about the results and do not try their best.



2.2 Gun time difference with age

From the data, we notice that there are many people who attend the competition more than one years. We want to see their gun time used with the age increasing.

First, we need to find the identity for each person. “name” can not be used singlely because many people have the same name. So we use the “age” and “year” to find the birthday year of each person. There may be some people who were born in the same year and have the same name. We choose name and year of birth together being id for each athlete. ‘hometown’ cannot be used directly, because the sometimes people use abbreviation, sometimes only use the state name.

Then, we use the id to find how many times every athlete attends the competition. We choose a subset in which every athlete attended the competition more than 6 times, there are 842 person and 6931 rows record their information.

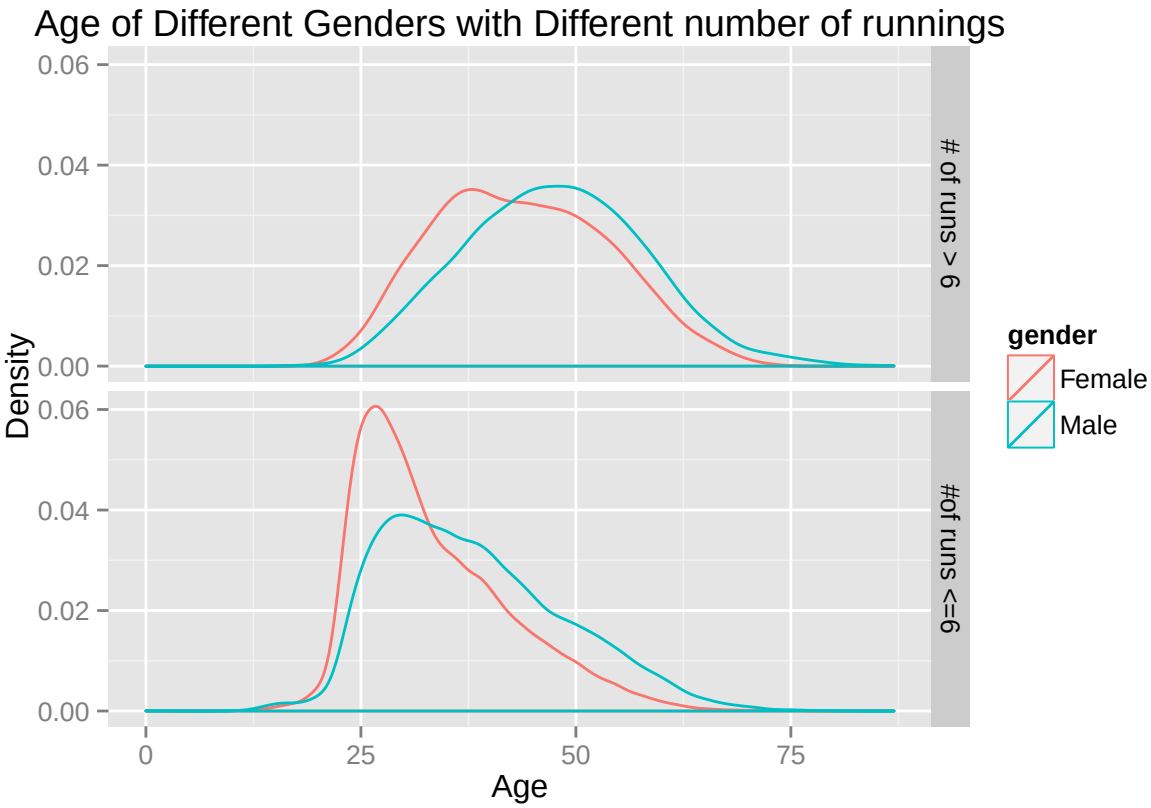
However, there are some error information we need to correct. For example, one person cannot attend the competition more than one times in the same year. There are seven id’s which have more than one record in the same year. There information(name, year of birth, the year with more than one records) is as follows.

```
## [1] "burt blackstone 1953 1999" "cara rooney 1980 2007"
## [3] "michael scott 1957 2002"   "patrick kunze 1980 2003"
```

Since there are only 4 id’s and 8 rows in this condition. I deleted these rows and choose records again to get people who attended the competition no less than five time. Then, add the first three characters of hometown into the identity of data and select athletes who run more than six times. This time there are 5146 observations which contain 629 athletes records.

2.2.1 Ages of athletes who take the competition different times

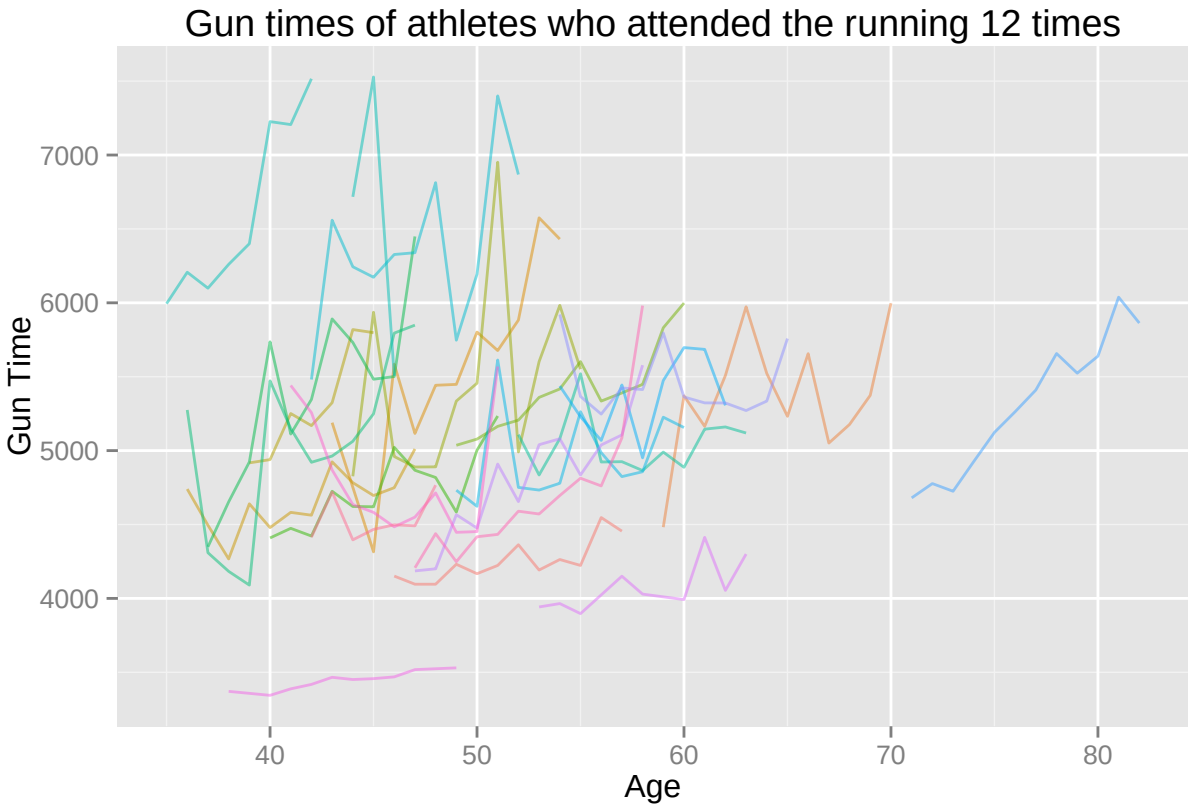
There is a interesting thing that most people take the competition several times are not young.



It seems that many people around 25 years did not persist to join the running every year. Instead, older people took the comtetition for more years. The next figure even show that most people take the running almost every year are more than 40 years old.

2.2.2 The influence of age on the guntime

First, we choose a subset of the data. There are 28 athletes attends the competition for 10 times. And their records are shown in the following figure.



We can see that although there are some variability, there is a trend that the gun time is increasing with the increase of age for almost all athletes.

We use linear regression to see the age effects(age fixed, id random) on all athletes who run more than 6 times. From the results as follows, we can see with the increase of age, their is an trends that gun time is also increased.

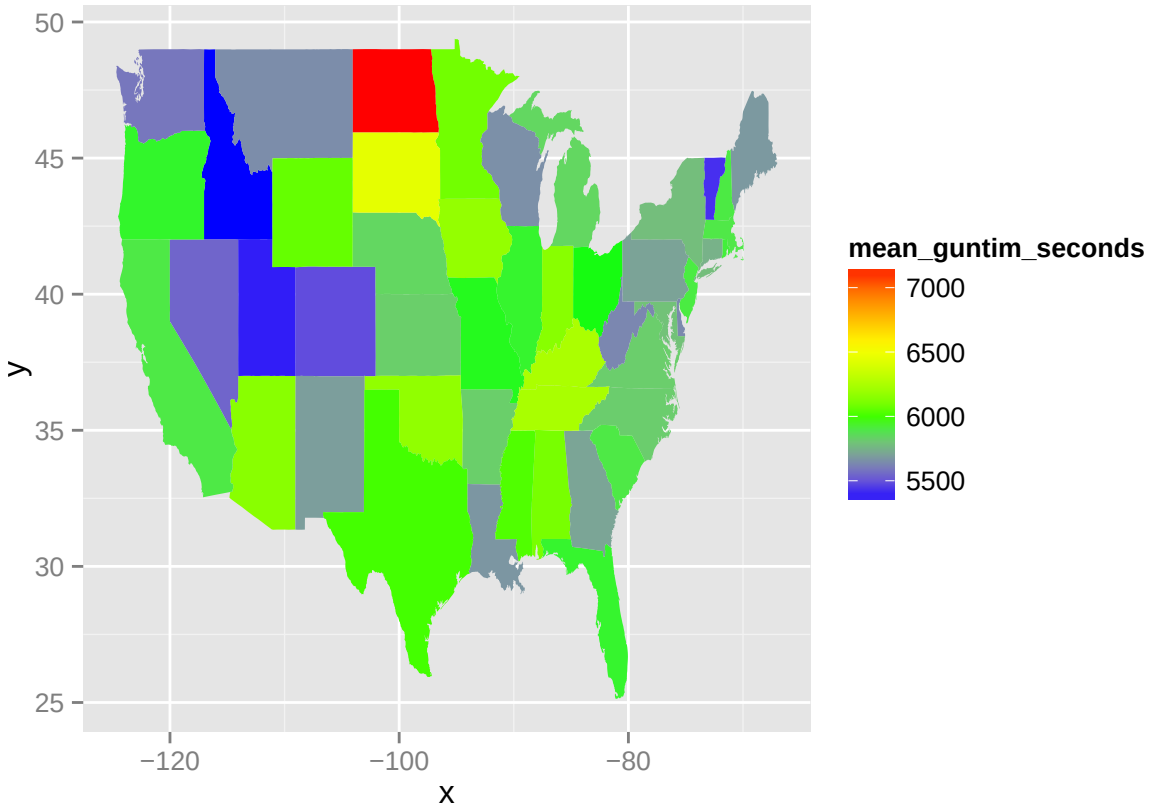
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: guntim ~ 1 + age + (1 | id2)
##    Data: five
##
## REML criterion at convergence: 79175.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2829 -0.5390 -0.1179  0.4123  8.1598
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id2      (Intercept) 804012   896.7
##  Residual             182890   427.7
## Number of obs: 5144, groups:  id2, 629
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 3093.348     93.005   33.26
## age           47.440      1.789   26.52
##
## Correlation of Fixed Effects:
```

```
##      (Intr)
## age -0.921
```

2.3 The Gun time in different areas.

2.3.1 The Gun time difference in USA

We first select athletes in USA. The methods in using the last part which in separated by blanks. Since the last part is always abbreviation of state names, we use 'maps' packages in R to match the last part of hometown to check which stands for states in USA. We can get 101997 rows. Although we may lose some records because their is no state abbreviation in hometown, there are enough samples. Then group the athletes whose hometown in America into diffenent states. Finally,count the mean of gun time of athletes in different states. The result is as follows:



We can see that in USA, people in Idaho run fastest. Then in Utah. People in North Dakota run slowest.

Next, check how many athletes in these 12 years in each state(count 2 is a athlete attend 2 times of the running).

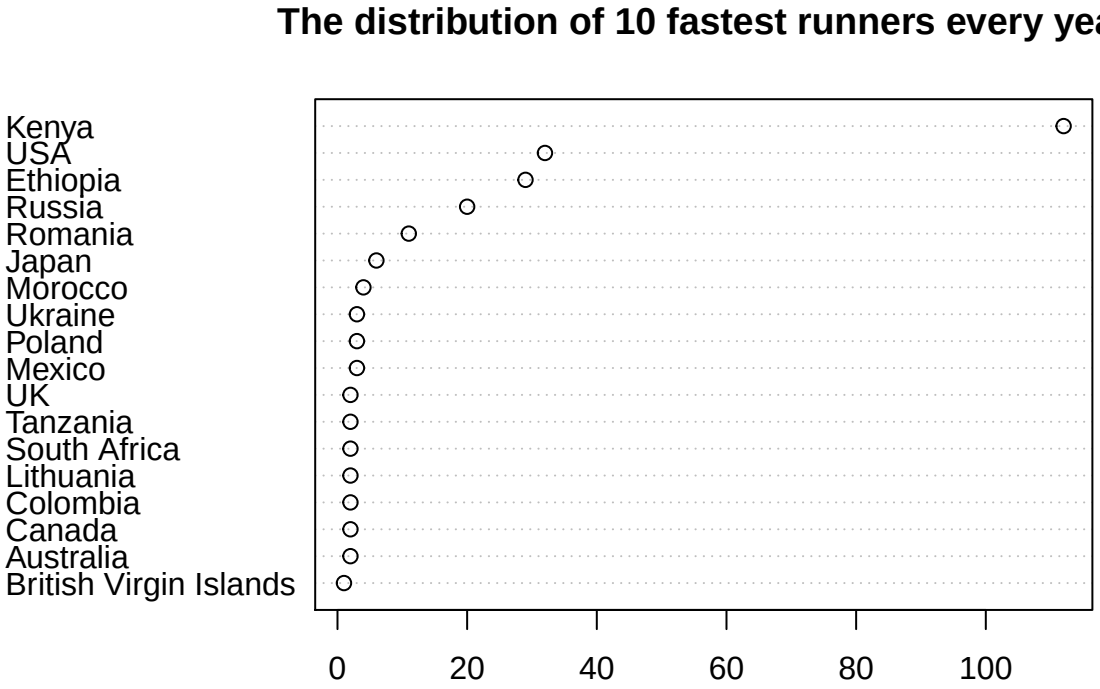
```
##
##    ak    al    ar    az    ca    co    ct    dc    de    fl    ga    hi
##    11    52    24    53   355   174   429 20230   297   383   263     4
##    ia    id    il    in    ks    ky    la    ma    md    me    mi    mn
##    49     5   394   106    77    53    23   767 27028   102   173   110
##    mo    ms    mt    nc    nd    ne    nh    nj    nm    nv    ny    oh
##    81    12     4   914     4    35   149  1396    42     9  3086   439
##    ok    or    pa    ri    sc    sd    tn    tx    ut    va    vt    wa
##    17    32  3320    87    53     7   104   312    23 40227    72    67
##    wi    wv    wy
```

```
##   116   214    13
```

We can see that this competition is popular only in three states, Virginia, Maryland and District of Columbia. So, may be the mean speed of athletes in each states cannot stands for the actual facts because there are not enough samples in other states.

2.3.2 People run fastest all over the world

We choose 20 fastest runners(10 males and 10 females) and their records each year. Then use 'maps' packages in R to get their countries. Then, we draw pie plot and world map to see their distribution.



We can see that many fastest runners are from Kenya. The race of people live in there may be better at running.

```
## 18 codes from your data successfully matched countries in the map
## 0 codes from your data failed to match with a country code in the map
## 226 codes from the map weren't represented in your data
```

fastest.athlete.numbers