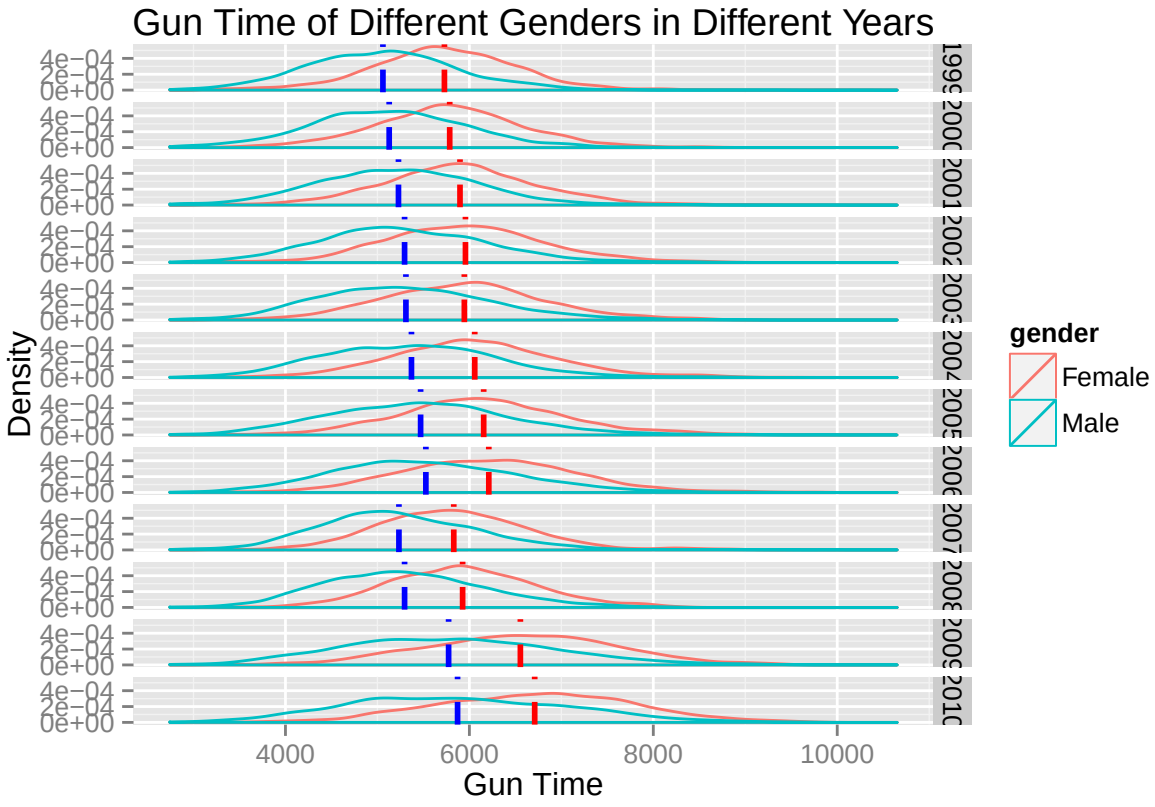
Appendix 2: Data Analysis

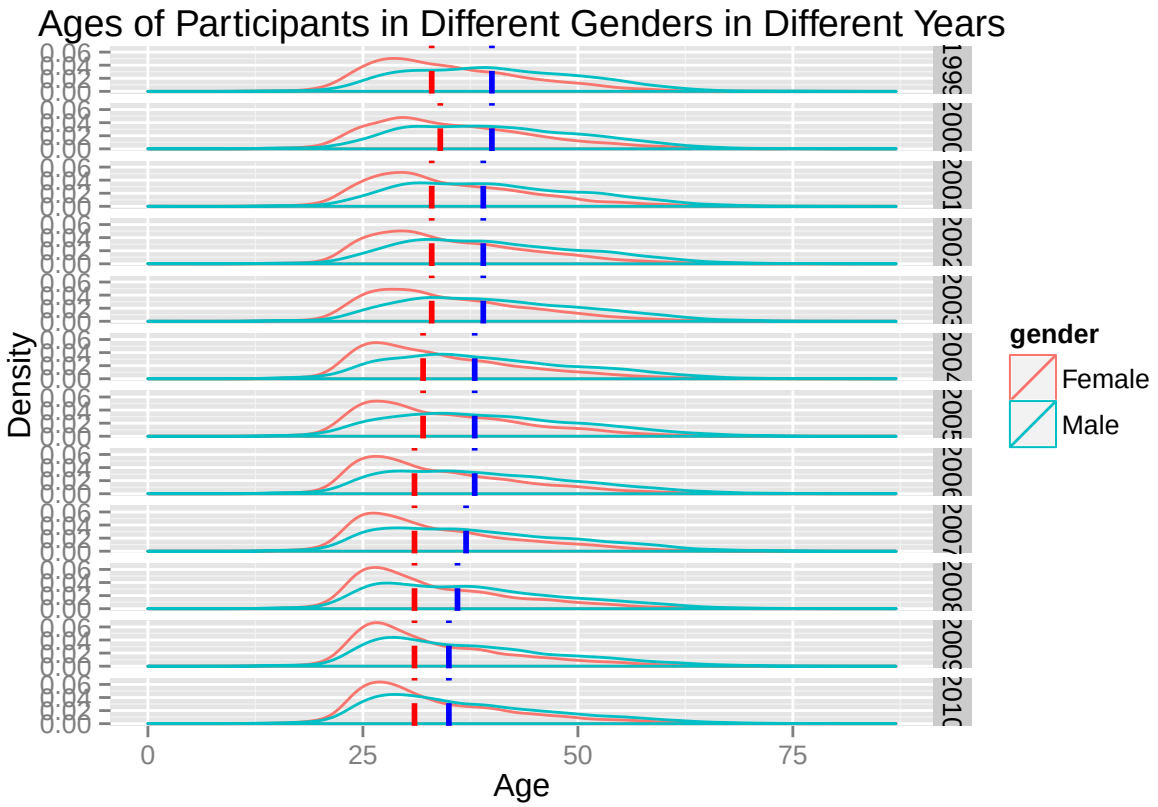
1. Gun time distribution of different genders in different years.

```
load('data.rda')
require(ggplot2)
require(plyr)
mean_gun_male = ddply(mileDat[mileDat$gender == 'Male', ], 'year',
                      summarise, gun.mean.male = mean(guntim, na.rm =T))
mean_gun_female = ddply(mileDat[mileDat$gender == 'Female',], 'year',
                        summarise, gun.mean.female = mean(guntim, na.rm =T))

ggplot(mileDat, aes(x = guntim, color = gender)) +
  geom_density(na.rm = T)  +
  facet_grid(year ~ .) +
  geom_vline(data = mean_gun_male, aes(xintercept = gun.mean.male),
             linetype = 'dashed', size = 1, colour = 'blue') +
  geom_vline(data = mean_gun_female, aes(xintercept = gun.mean.female),
             linetype = 'dashed', size = 1, colour = 'red') +
  xlab("Gun Time") + ylab("Density") +
  ggtitle("Gun Time of Different Genders in Different Years")
```




```
# Ages of Participants in Different Genders in Different Years
median_age_male = ddply(mileDat[mileDat$gender == 'Male', ], 'year',
                        summarise, age.median.male = median(age, na.rm =T))
```

```
median_age_female = ddply(mileDat[mileDat$gender == 'Female',], 'year',
                          summarise, age.median.female = median(age, na.rm =T))

ggplot(mileDat, aes(x = age, color = gender)) +
  geom_density(na.rm = T)  +
  facet_grid(year ~ .) +
  geom_vline(data = median_age_male, aes(xintercept = age.median.male),
             linetype = 'dashed', size = 1, colour = 'blue') +
  geom_vline(data = median_age_female, aes(xintercept = age.median.female),
             linetype = 'dashed', size = 1, colour = 'red') +  xlab("Age") + ylab("Density") +
  ggtitle("Ages of Participants in Different Genders in Different Years")
```



2. Gun time with age

2.1 The influence of age on the guntime

```
mileDat$yearofbirth = as.integer(as.character(mileDat$year)) - mileDat$age
#use name and birth year together to be id
mileDat$id = paste(mileDat$name, mileDat$yearofbirth)
mileDat$idWithYear = paste(mileDat$id, mileDat$year)

#number of runs of each athlete
nruns = aggregate(mileDat$yearofbirth, by=list(who = mileDat$id), length)
goo = merge(mileDat, nruns, by.x = 'id',by.y = 'who')
```

```r
five = subset(goo, x>6) #athletes run more than 6 times
#conflicts, one person cannot run more than one times in one year
z = names(which(table(five$idWithYear) == 2)); z
```

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

```r
#delete the records of althletes which has the conflicts
x = sapply(1:length(z), function(i) {
  tmp = nchar(z[i])
  substr(z[i], 1, tmp-5)
})
five= five[!(five$id %in% x),]
nruns = aggregate(five$yearofbirth, by=list(who = five$id), length)
goo = merge(five, nruns, by.x = 'id',by.y = 'who')
five = subset(goo, x.y >6)
table(table(five$id))
```

```
##
##   7   8   9  10  11  12
## 331 221 142  72  48  24
```

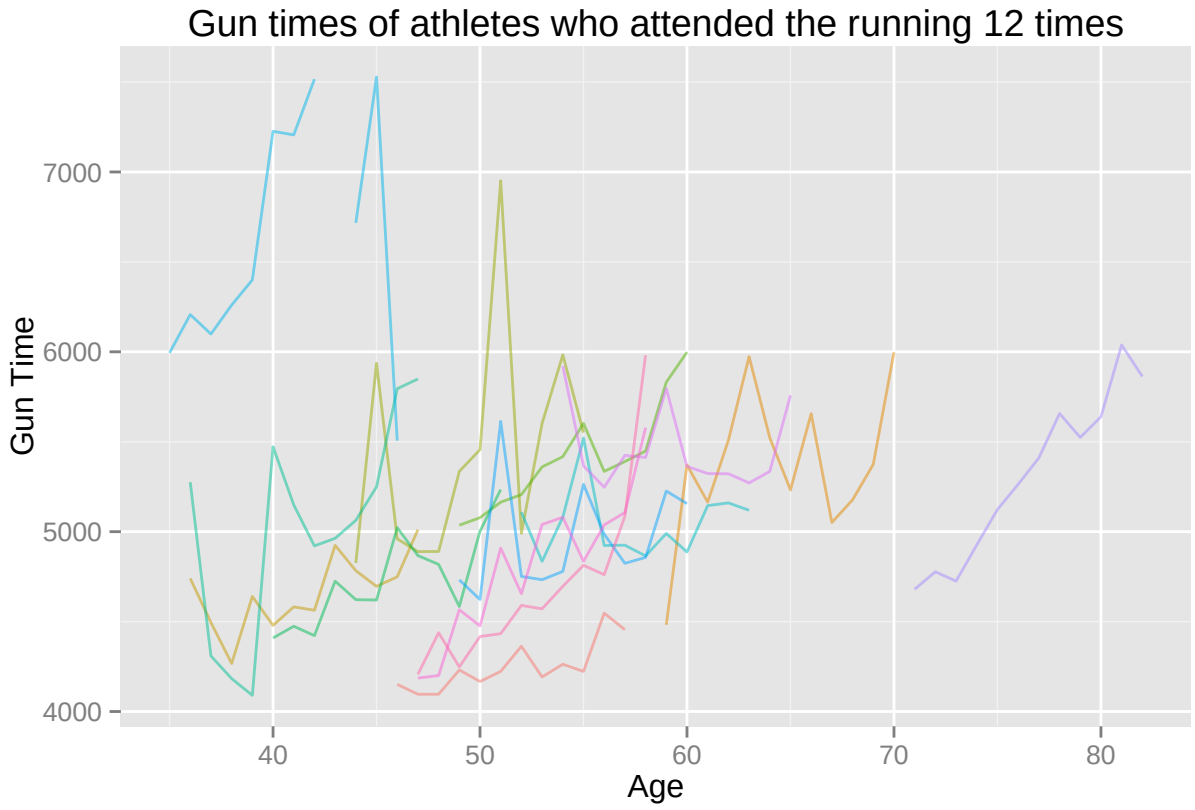
```r
#Add the first 3 characters into id can clean the data again
five$id2 = paste(five$id, substr(five$hometown,1,3))
nruns = aggregate(five$yearofbirth, by=list(who = five$id2), length)
goo = merge(five, nruns, by.x = 'id2',by.y = 'who')
five = subset(goo, x >6)
table(table(five$id2))
```

```
##
##   7   8   9  10  11  12
## 255 163 113  52  32  14
```

```r
#number of athlets who run more than 6 times
sum(table(table(five$id2)))
```

```
## [1] 629
```

```r
#There are 14 athletes who take the running every year
sub1 = five[five$x == 12,]
sub1 = sub1[with(sub1, order(id2, age)),]
ggplot(sub1, aes(age,guntim, group = id,colour = id)) + geom_path(alpha = 0.5) +
  theme(legend.position = "none") +
  xlab("Age") + ylab("Gun Time") +
  ggtitle("Gun times of athletes who attended the running 12 times")
```




```
#linear regression, age fixed, id random
library(lme4)
```

```
## Loading required package: Matrix
## Loading required package: Rcpp
```

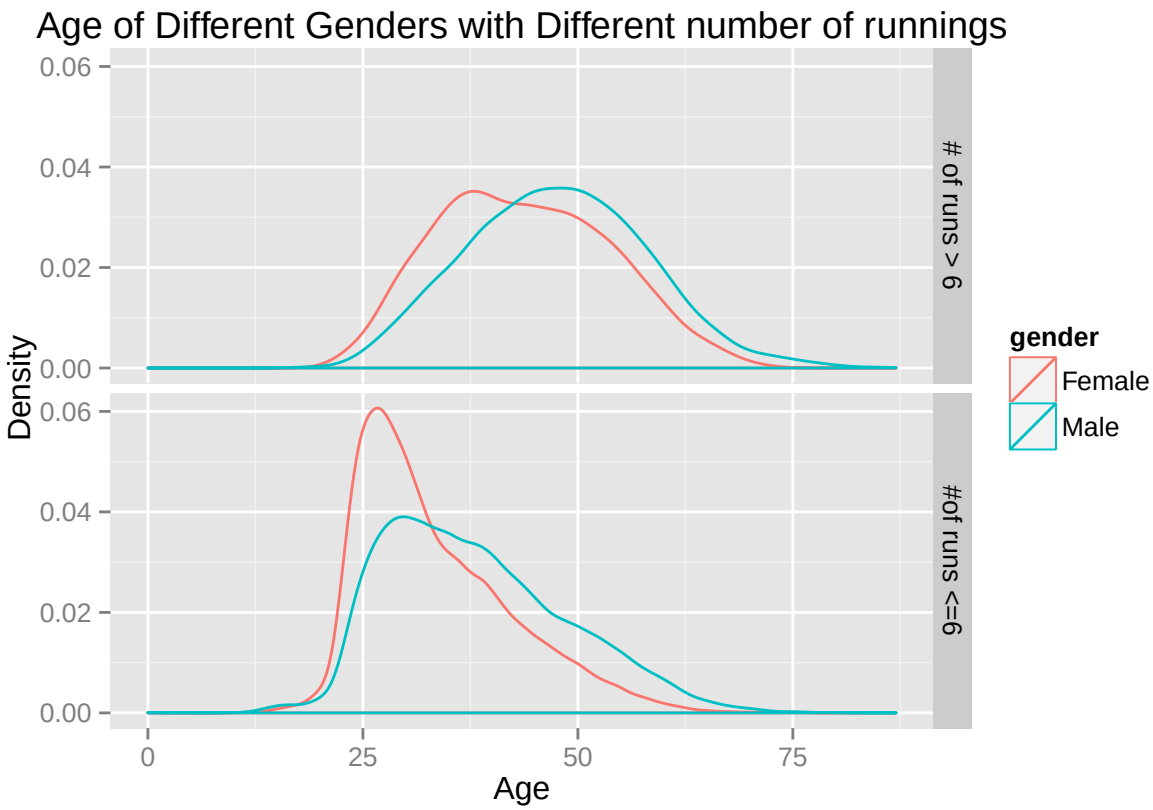
```
model = lmer(guntim ~ 1 + age + (1|id2), data = five)
summary(model)
```

```
## 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
```

2.2 Ages of athletes who take the competition different times

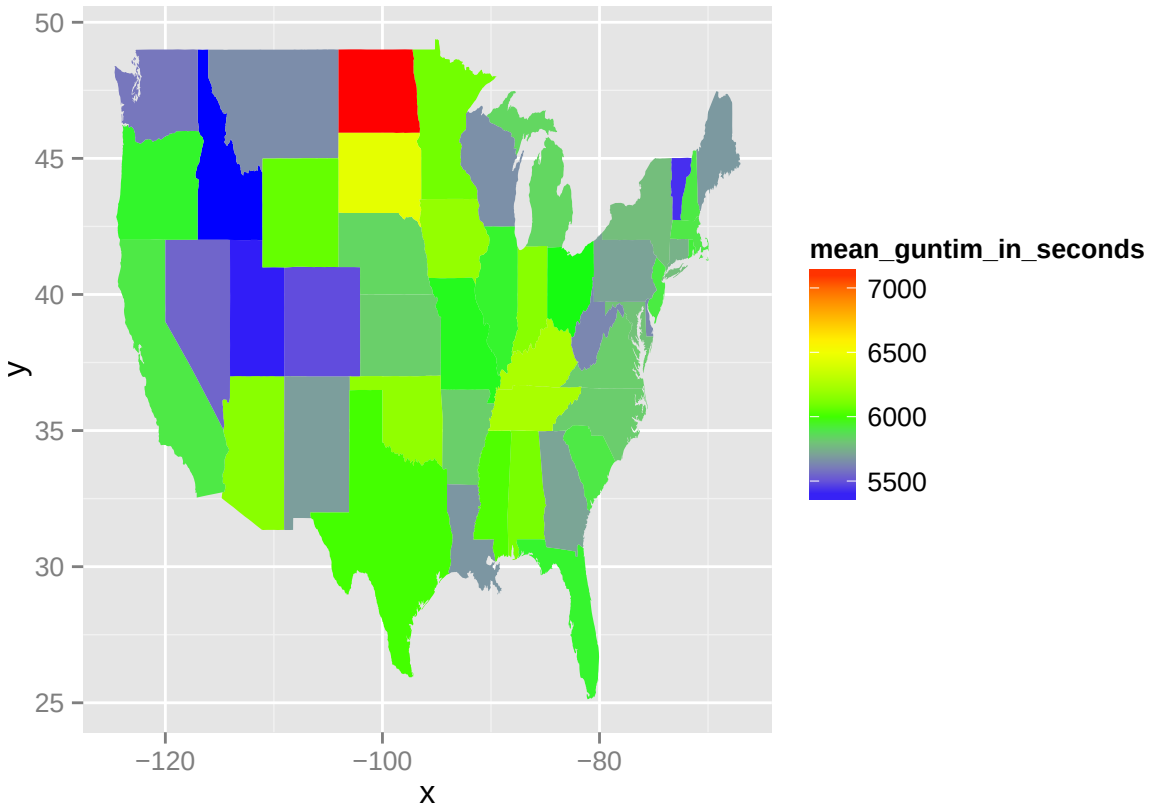
```
nruns = aggregate(mileDat$yearofbirth, by=list(who = mileDat$id), length)
goo = merge(mileDat, nruns, by.x = 'id',by.y = 'who')
goo$runtime = sapply(goo$x, function(x) {
  if (x <=6){
    z = "#of runs <=6"
  }else{
    z = "# of runs > 6"
  }
})

ggplot(goo, aes(x = age, color = gender)) +
  geom_density(na.rm = T)  +
  facet_grid(runtime ~ .) +
  xlab("Age") + ylab("Density") +
  ggtitle("Age of Different Genders with Different number of runnings")
```



3. The Gun time in different areas

3.1 The Gun time difference in USA

```
library(maps)
data(world.cities)
data(us.cities)
#all countries in the world
countries = unique(world.cities$country.etc)
#all states in USA(abbreviation)
states = unique(us.cities$country.etc)

#the last part in hometown(splited by blanks)
state = sapply(mileDat$hometown, function(i) {
  tmp = unlist(strsplit(i,'\\s'))
  tmp = tmp[length(tmp)]
})

#Find hometown belongs to USA and out of USA
nusa = match(state, tolower(states))
outusa = which(is.na(nusa))
inusa = which(!is.na(nusa))

#Add state information to each row if the hometown belongs to USA
mileDat$state = rep(NA, dim(mileDat)[1])
mileDat$state[inusa] = state[inusa]


#the mean guntime of each state
mean_gun = ddply(mileDat[!is.na(mileDat$state),], 'state',
                 summarise, state.mean = mean(guntim, na.rm =T))

data(state.fips)

wholename_state = sapply(as.character(state.fips$polyname), function(i){
  unlist(strsplit(i, ':'))[1]
})
wholename_state = unique(wholename_state)

#the mean gun time of each state
res = data.frame(name = wholename_state,
                 mean_guntim_in_seconds = rep(0,length(wholename_state)))
for(i in 1: length(wholename_state)){
  m = grep(res$name[i], state.fips$polyname)[1]
  res$mean_guntim_in_seconds[i] = mean_gun$state.mean[mean_gun$state == tolower(state.fips$abb[m])]
}

states_map <- map_data("state")
ggplot(res, aes(map_id = name)) +
    geom_map(aes(fill = mean_guntim_in_seconds), map = states_map) +
    scale_fill_gradientn(colours=c("blue","green","yellow","red")) +
    expand_limits(x = states_map$long, y = states_map$lat)
```

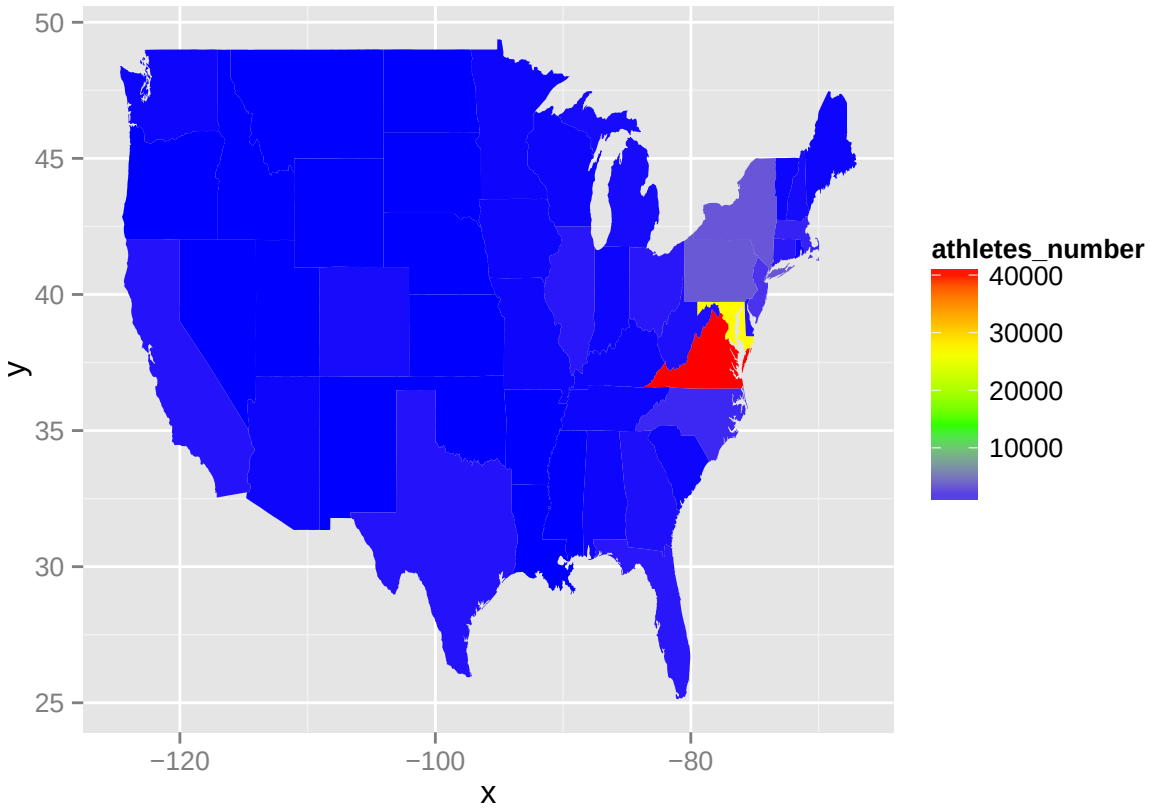



```
numathletes = ddply(mileDat[!is.na(mileDat$state),], 'state',
                    summarise, athlete.number = length(guntim))

#number of athletes in each states(with whole name of each state)
y = data.frame(name = wholename_state,
               athletes_number = rep(0,length(wholename_state)))
for(i in 1: length(wholename_state)){
  m = grep(y$name[i], state.fips$polyname)[1]
  y$athletes_number[i] = numathletes$athlete.number[numathletes$state == tolower(state.fips$abb[m])]
}
y
```

```
##                    name athletes_number
## 1               alabama              52
## 2               arizona              53
## 3              arkansas              24
## 4            california             355
## 5              colorado             174
## 6           connecticut             429
## 7              delaware             297
## 8  district of columbia           20230
## 9               florida             383
## 10              georgia             263
## 11                idaho               5
## 12             illinois             394
## 13              indiana             106
## 14                 iowa              49
```

```
## 15              kansas             24
## 16            kentucky             53
## 17           louisiana             23
## 18               maine            102
## 19            maryland          27028
## 20       massachusetts            767
## 21            michigan            173
## 22           minnesota            110
## 23         mississippi             12
## 24            missouri             81
## 25             montana              4
## 26            nebraska             35
## 27              nevada              9
## 28       new hampshire            149
## 29          new jersey           1396
## 30          new mexico             42
## 31            new york           3086
## 32      north carolina            914
## 33        north dakota              4
## 34                ohio            439
## 35            oklahoma             17
## 36              oregon             32
## 37        pennsylvania           3320
## 38        rhode island             87
## 39      south carolina             53
## 40        south dakota              7
## 41           tennessee            104
## 42               texas            312
## 43                utah             23
## 44             vermont             72
## 45            virginia          40227
## 46          washington             67
## 47       west virginia            214
## 48           wisconsin            116
## 49             wyoming             13
```

```
states_map <- map_data("state")
ggplot(y, aes(map_id = name)) +
    geom_map(aes(fill = athletes_number), map = states_map) +
    scale_fill_gradientn(colours=c("blue","green","yellow","red")) +
    expand_limits(x = states_map$long, y = states_map$lat)
```

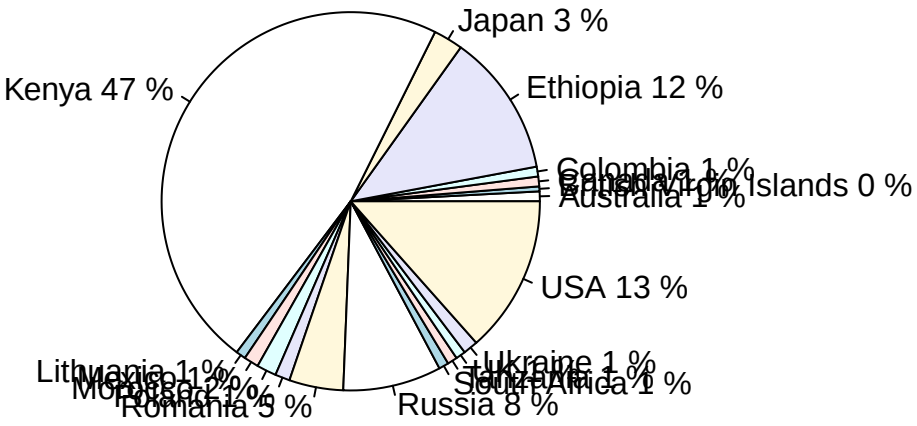


3.2 People run fastest all over the world

```
#hometown of 20 fastest runners(10 female, 10 male) each year
home = mileDat$hometown[mileDat$place <= 10 & !is.na(mileDat$hometown)]

#match hometown to different countries
#since the records in hometown always not wholenames, we use it as predix to match countries
b = paste('^', home, sep = '')
z = sapply(b, function(i) {
  grep(i, tolower(countries))[1]
})

#change the name of countries to uniform country names
res = sapply(1:length(home),function(i) {
  countries[z[i]]
})

#There are 32 hometown which we do not know which country they belong to
#see if they belongs to America
nomatch = which(is.na(res))
subhometown = home[nomatch]
substate = sapply(subhometown, function(i) {
  tmp = unlist(strsplit(i,'\\s'))
  tmp = tmp[length(tmp)]
})
ifusa = match(substate, tolower(states))
```

```
res[nomatch[which(!is.na(ifusa))]] = 'USA'
res[home == "united states"] = 'USA'
res[home == "united kingdom"] = 'UK'
res[c(49, 140)] = 'South Africa'
res[c(197,198)] = 'USA'
res = res[!is.na(res)]


pieplot = function(x,nam)
{
  n = length(x)
  z = table(x)
  a = names(z)
  p = round(100 * z / n)
  lab = paste(a, p)
  lab = paste(lab, '%', sep = " ")
  pie(table(x), labels = lab, main = nam)
}

pieplot(res, "The distribution of 10 fastest runners every year")
```

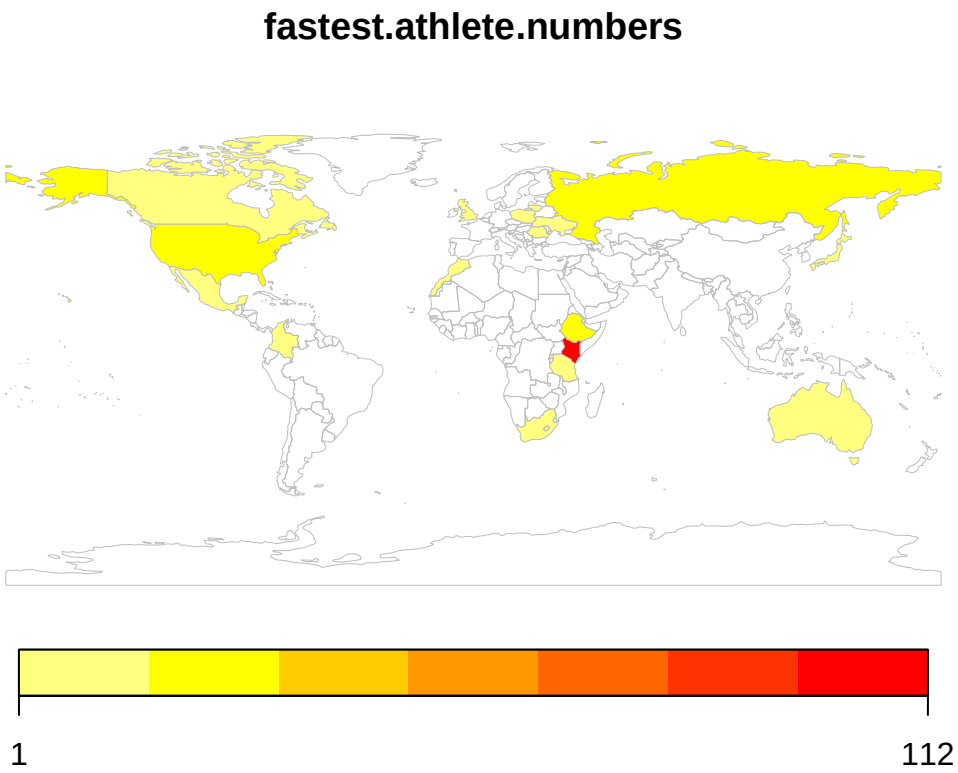
The distribution of 10 fastest runners every year




```
first10 = data.frame(country = res)

x = ddply(first10, 'country', summarise, fastest.athlete.numbers = length(country))
library(rworldmap)

spdf <- joinCountryData2Map(x, joinCode="NAME", nameJoinColumn="country")
```

```
## 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
```

```
mapCountryData(spdf, nameColumnToPlot="fastest.athlete.numbers", catMethod="fixedWidth")
```




```
nathlete = table(res)

nathlete = nathlete[order(nathlete, decreasing = F)]
dotchart(nathlete, main ='The distribution of 10 fastest runners every year')
```