STA242
Project 1

999452701
Junxiao Bu

Data Description

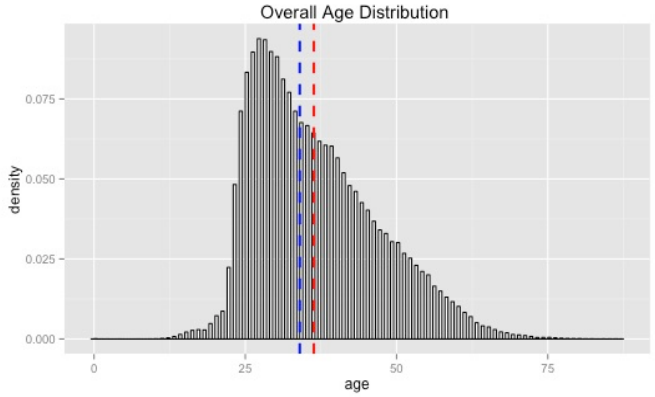
After data munging and combing all the observations into one dataframe. There are **113190** observations in total. Eight variables in the each file were kept or saved in this file. Their names are: “place”, “div/total”, “name”, “hometown”, “age”, “gun time”, “net time” and “time”. In addition to the conversion of character strings to numeric, I also create two new variables, year and sex.

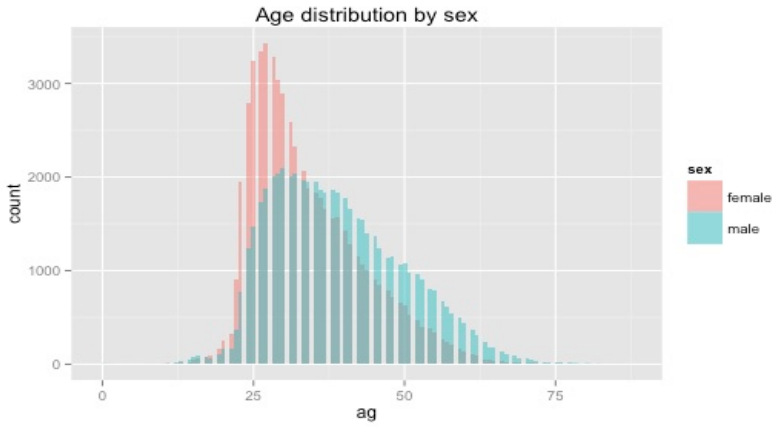
Besides, in order to calculate each runner’s time, one new variable called “race_time” is created. For those years having information of gun time, gun time will be used as race_time. For those years without having information of gun time, “time” will be used as race_time. All the variables related to time are converted into numeric values. The unit of time is minute.

For analysis part, I mainly discover two variables: age and race_time. Their distributions and relation between two variables will be analyzed. Each year’s best record players and their characteristics will be analyzed. Besides, a record for an individual runner across years will also be created. One unique ID called “id_birth” will be created. Some analysis such as who attend games at least ten times will be illustrated.

PART1: Age Distribution

The overall distribution and distribution by sex of age are:

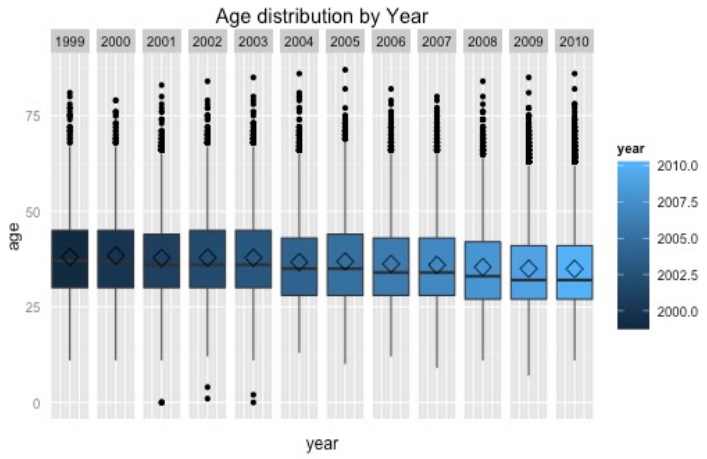


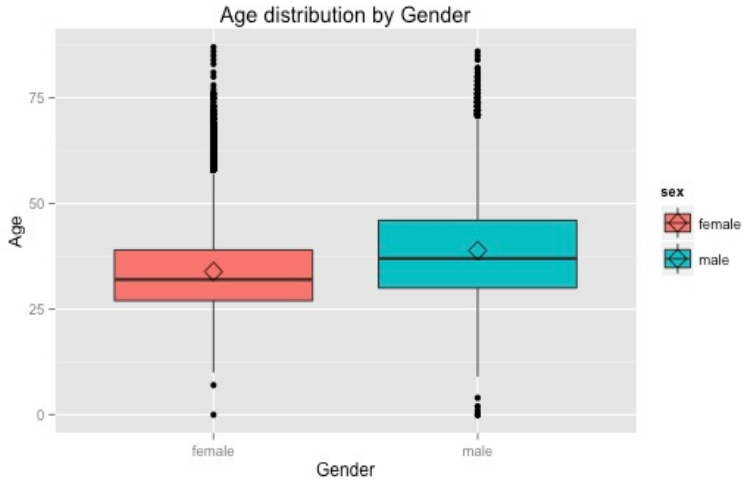


The overall distribution and distribution by sex show the same pattern. The distributions are slightly right-skewed. Besides, the plots show that there are more younger women runners than men runners, which is also proven by the following table. The median age of women runners are much smaller than the median age of men runners.

Gender	Mean of Age	Median of Age
Female	33.865	32
Male	38.857	37

Boxplots to illustrate the age distribution by year and gender are shown below. Generally, women runners' median ages are smaller than men runners'. These side-by-side boxplots of age for each race year show a reasonable age distribution. For example, the lower quartile for all years ranges between 29 and 32.



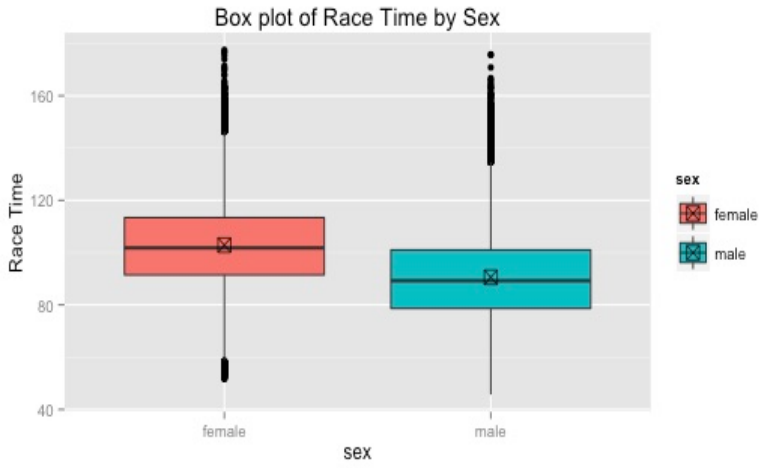


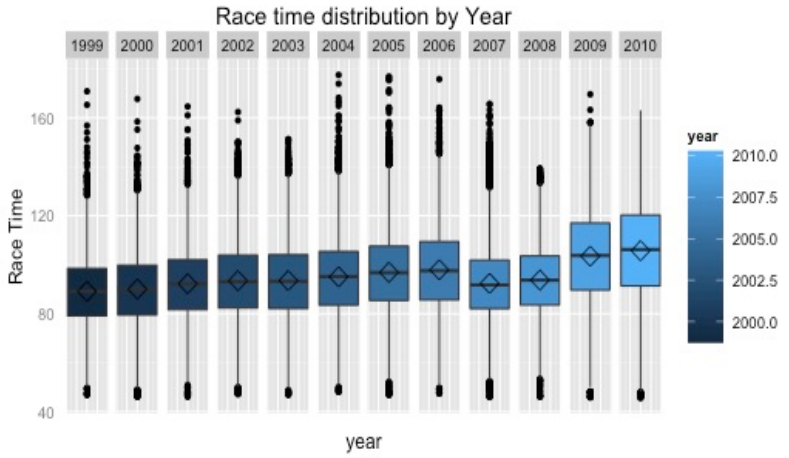
PART2: Race Time Distribution

Summary table of race time by sex:

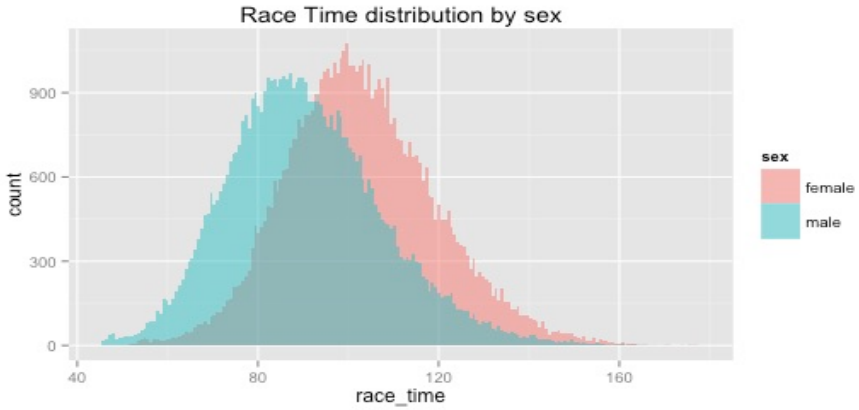
Sex	Mean of Race Time	Median of Race Time
Female	102.851	101.8
Male	90.549	89.233

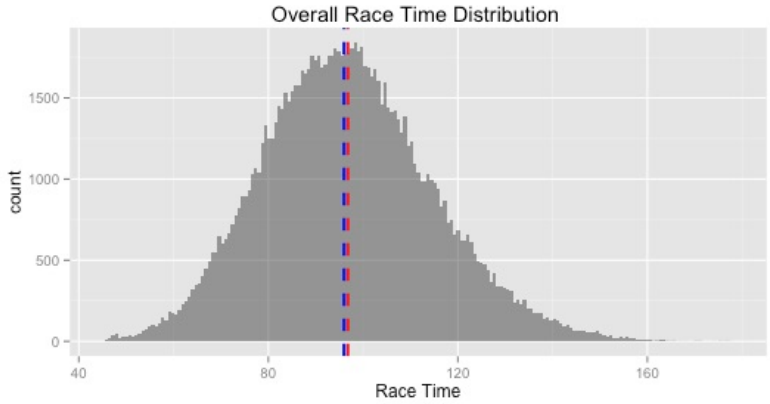
From the summary table, the mean and median of race time for female are longer than male's. This is not surprising since men and women have different physical conditions. The boxplots by sex and year are shown below. These side-by-side boxplots of race time for each race year show a reasonable race distribution. For example, the lower quartile for all years ranges between 90 and 110.



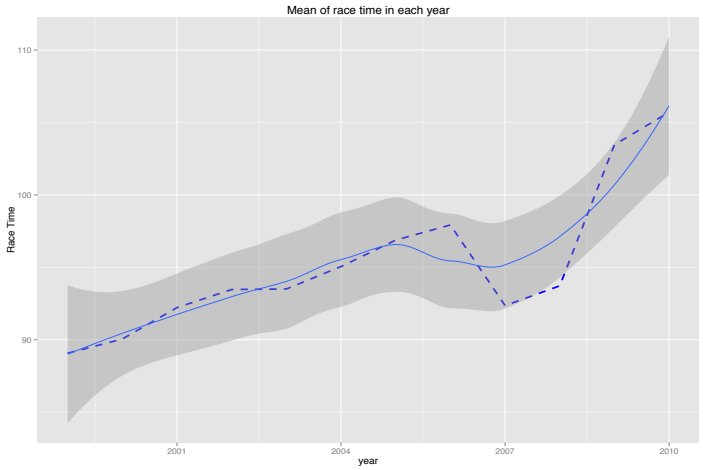


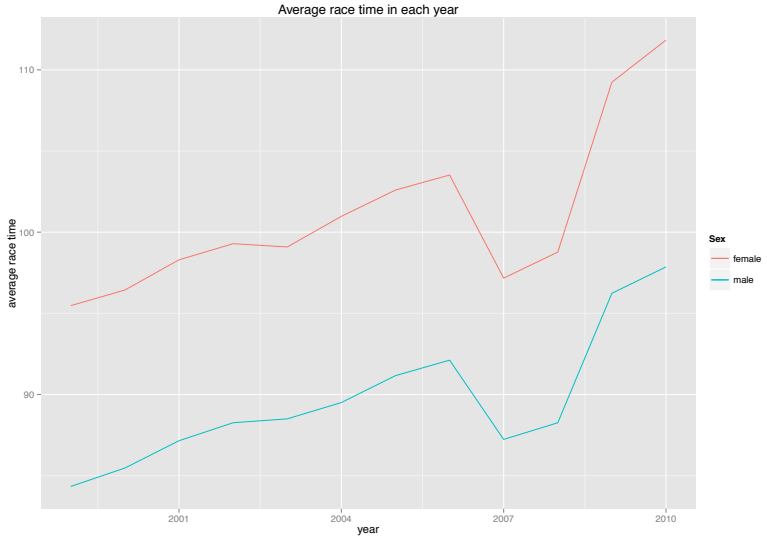
The overall distribution and distribution by sex show the same pattern. The distributions are reasonably symmetric. The plots also show that men's distribution is shifted to the left of women's distribution, which is also proven above—men's mean time is lower than women's.



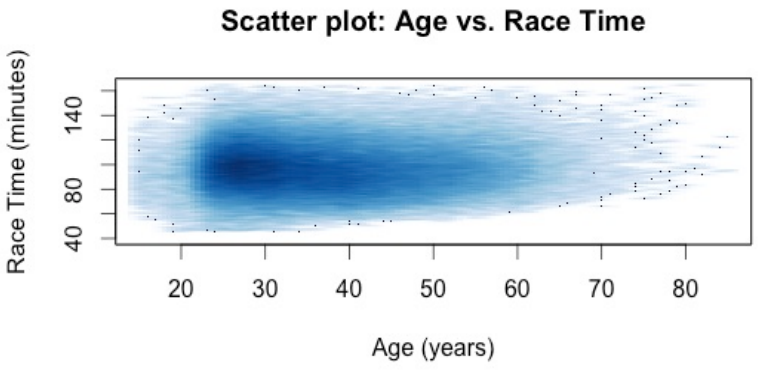


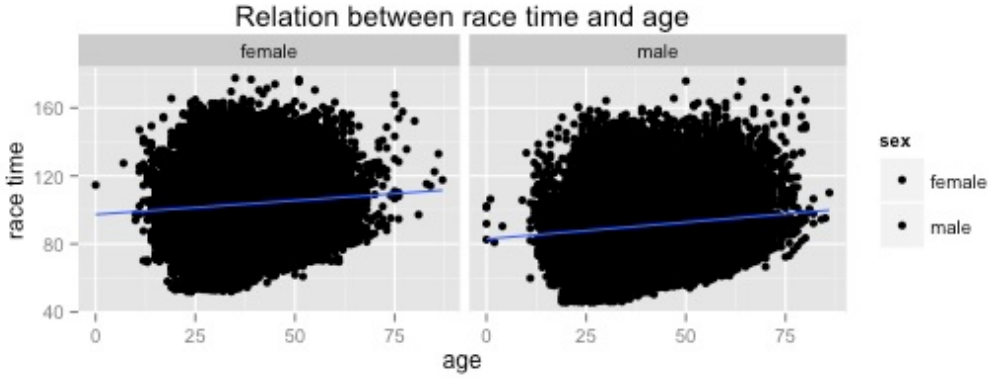
From the overall average race time and average race time by year, one interesting trend is observed. The average time is increasing. Since the average ages of runners are also increasing. One assumption of relation between race time and ages can be explored.





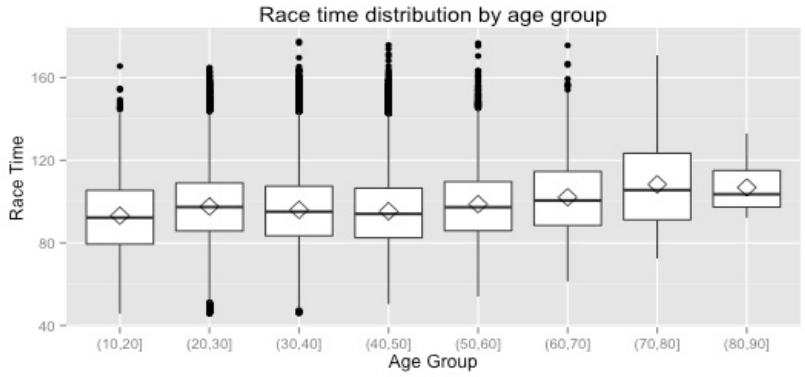
The scatter plot between race time and ages are shown below:





Now we see the shape of the high-density region containing most of the runners and the slight upward trend of time with increasing age. The scatter plots by gender also show the slight upward trend. By adding the simple linear regression line between age and race time, age has positive relation with race time, which means age, has negative relation with runners' speed. Runners become slower as their ages grow.

Now all the runners are separated into different group by their ages. The side-by-side boxplots of runners' race time vs. age groups is shown below. As age increases, all the quartiles increase. However, the box becomes asymmetrical with age, which indicates that the upper quartile increases faster than the median and lower quartile.



PART 3: Explore each year's best record player's identity information

QUESTION 1: Who are the domestic winners in each year? Where are they from? What are their records?

Year	Name	Hometown	Race Time	Gender
1999	Eric Morrison	Lakewood CO	50.58	Male
2000	Jimmy Hearld	Louisville KY	47.98	Male
2001	Christopher Graff	Washington DC	47.36	Male
2002	Jason Dejoy	Randallstown MD	49.75	Male
2003	Fred Kieser	Cleveland OH	50.46	Male
2004	Matthew Wagoner	New Cumberland PA	51.25	Male
2005	Michael Wardian	Arlington VA	52	Male
2006	Patrick MacAdie	Washington DC	50.32	Male
2007	Stephen Meinelt	Washington DC	48.75	Male
2008	Steven Crane	Silver Spring MD	49.68	Male
2009	Jason Hartman	Boulder CO	48.05	Male
2010	Tim Young	Fredericksburg VA	49.07	Male

Not surprising, all the best domestic players are male. One interesting result is that their hometowns are all located in east part of the U.S. One possible reason is that this race game holds in Washington DC. People from central and west parts of the country are less interested in attending this race game.

QUESTION 2: Who is each year's top international player? Where are they from? What are their records?

The full report is in the following table:

Year	Name	Hometown	Race Time
1999	Worku Bikila	Ethiopia	46.98
2000	Reuben Cheruiyot	Kenya	46.12
2001	John Korir	Kenya	46.2
2002	Rueben Cheruiyot	Kenya	47.22
2003	John Korir	Kenya	46.93
2004	Nelson Kiplagat	Kenya	48.2
2005	John K Korir	Kenya	46.93
2006	Gilbert Okari	Kenya	47.42
2007	Tadesse Tola	Ethiopia	46.02
2008	Ridouane Harroufi	Morocco	46.23
2009	Ridouane Harroufi	Morocco	45.93
2010	Stephen Tum	Kenya	45.72

The table shows the interesting result that the best international runners in each year are all from African countries. This is not so surprising since African players are expert in long-distance race.

PART 4: Record for an Individual Runner across Years

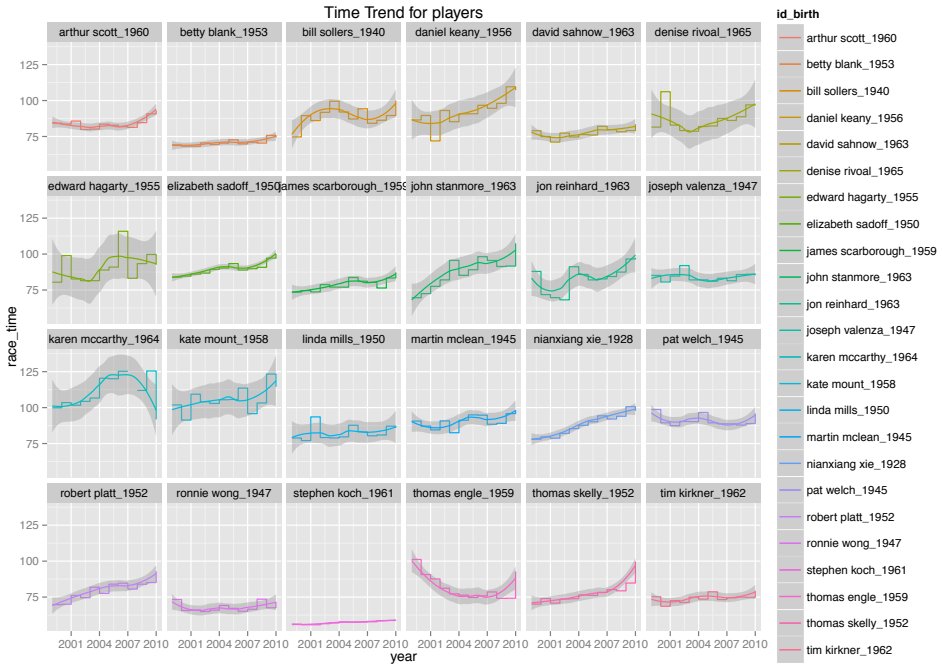
Name variable and hometown variable are cleaned in order to eliminate space and irrelevant symbol. There are **113190** entrants for 12 years' race results. Player's names' appearance times across years are in the following table.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
50600	12425	4546	2097	1092	551	334	221	144	73	48	24

We see that over 13000 names appear 2 times throughout the 12 races. One name appears 26 times, and we know this name must correspond to at least 2 people because we have only 12 years of race results. In order to classify that which players have attended games in several years, an ID needs to be created. Firstly, we use each player's year of attendance minus the corresponding age as each player's approximately date of birth. Then each player's name and year of birth would be pasted together as the new ID: **id_birth**. There are **72155** unique IDs in this case. Since our goal is to study how an athlete's time changes with age, let's focus on those IDs that appear in at least 10 times. After some research, I found that there are **73** players, who have attended the race game at least ten times. Their names and year of birth are in the following table. We can also explore people who have attended the game each year (12 times). I found there are **24** people who attended the race game 12 times.

Arthur Scott_1960	Martin Mclean_1945
Betty Blank_1953	Bill Sollers_1940
Karen Mccarthy_1964	Thomas Engle_1959
Daniel Keany_1956	David Sahnow_1963
Denise Rivoal_1965	Edward Hagarty_1955
Elizabeth Sadoff_1950	Tim Kirkner_1962
James Scarborough_1959	John Stanmore_1963
Jon Reinhard_1963	Joseph Valenza_1947
Kate Mount_1958	Linda Mills_1950
Nianxiang Xie_1928	Pat Welch_1945
Thomas Skelly_1952	Robert Platt_1952
Ronnie Wong_1947	Stephen Koch_1961

The plot below can analyze the race time trend for these 24 players.



From the plot, I can find that there is an no so obvious trend that each player's race time is increasing along with the year grows. Some players' changes are quick and some's are not.

Conclusion

In this dataset, we primarily explore the distribution of runners' age and their corresponding race time. Not surprising, their race times are longer when the players' ages are increasing. Women runners' race time are longer than male runners, but their ages are smaller in average. For each year's records, all the champions are from African countries since they are expert in long-distance race game. For each individual's specific race performance, not surprising, their race time are longer along with the times they have attended the race game.

Reference:

1. Lecture Notes
2. Piazza posts: 31,43,54,95

APPENDIX 1: Source code of cleaning data and combine them into a complete dataframe.

```
## This part generates the complete dataframe.

#### two global variable:

## colname: columns I want to keep for further analysis

colname = c("place","div/tot","name","hometown","ag","gun",

          "net","time")

## paths: generate all the filenames

paths = list.files("~/Desktop/STA242/homework1/data")


# There are 4 main functions in this project. In each main function,

# several short functions would be called and they are listed in the

# end.

###############  Step1: use read.fwf to read one file ########################

## This function will use read.fwf function to read in all the files

Readin_one_table = function(filename){

  ## use "process_file" function to delete space line,

  ## footnote and *# symbol

  ## save the new file named "file_adj"

  process_file(filename)

  file = readLines("file_adj")

  ### special case for women2001: no "==" line

  if(grepl("women10Mile_2001",filename)){

    ## the width of women2001 is the same as men2001. use men2001

    ## width as width for women2001.

    index2001 = grep("- WOMEN",file)

    width = width_get(readLines("men10Mile_2001"),index2001+12)
```

```
   raw_data = read.fwf("file_adj",widths=width,skip=index2001+1)
   names(raw_data) = c("place","num","name","ag","hometown","net","gun")
  }
  else{
     index_equ = line_skip(file)
      ## obtain the width to partition all the columns for each variable
      width = width_get(file,index_equ)
      raw_data = read.fwf("file_adj",widths=width,skip=index_equ-2)
      col_name = tolower(unlist(raw_data[1,]))
     ## get rid of all the blank space
      col_name = gsub(" {1,}","",col_name)
      names(raw_data) = col_name
     ## first two lines are headers and "="
      raw_data = raw_data[-c(1,2),]
  }
  raw_data
}
##########################################


################step2:formatting the dataframe from step 1################
###############Goal:1.keep columns we need;2.process special cases:
### 2006 split two columns; 2001: without female header
format_table = function(data,colname,filename){
  ## special case for women2001:same format as men2001,use men2001's format
  if (grepl("women10Mile_2001",filename)){
    ## for women2001,remove "num" columns,create four new columns:
    ## div/tot,year,time and sex
      men_sub = cbind(data[,-2],"div/tot" = rep(NA,nrow(data)),
```

```
            time=rep(NA,nrow(data)),year=rep(2001,nrow(data)),
             sex=rep("female",nrow(data)))
}
else{
   process_file(filename)
   file = readLines("file_adj")
   ## index to find the header line
   index = line_skip(file)
   ## get the header line
   header = tolower(file[index-1])
   header = gsub(" ","",header)
  ## find which columns we want are in the dataframe.
  head_split = match_header(header,colname)
  ## column are in the dataframe
  head_yes = head_split[[1]]
  ## column names are not in the dataframe
  head_no = head_split[[2]]
  ## subset the original dataset,which contains columns we want already
  var_name = data_sub(head_split,data)
  men_sub = data[,which(colnames(data) %in% var_name)]
  ## columnbind columns doesn't exist in original dataset, but we want in
  ## colname vector.we set those columns with values NA.
  men_sub = bind_data(head_no,men_sub)
  ## generate year and gender columns
  men_sub = year_sex(men_sub,filename)
  ## split the special case of 2006:hometown and netime in one column
  if(grepl("(^women|^men)10Mile_2006",filename)){
    men_sub = split_home_nettime(men_sub)
```

```
    }
  }
  men_sub
}
################################################################################

################### step3: get one formatted table ############################
#####combine step1 and step2. Format the column names of one table
get_one_table = function(filename){
  colname = c("place","div/tot","name","hometown","ag","gun",
              "net","time")
  ## readin the raw data with all columns
  raw_data = Readin_one_table(filename)
  ## subset the data with columns we want
  formatted_data = format_table(raw_data,colname,filename)
  ## modify columns names to standard format
  names(formatted_data) = gsub("[[:space:]]", "", names(formatted_data))
  names(formatted_data)[names(formatted_data) == 'guntim'] = "gun"
  names(formatted_data)[names(formatted_data) == 'nettim'] = "net"
  names(formatted_data)[names(formatted_data) == 'netti'] = "net"
  ## change variable names to the same order
  formatted_data = formatted_data[c(colname,"sex","year")]
}
################################################

########## step4: loop over all the filenames to get the final dataframe######
Get_all_table = function(paths){
  data_final = data.frame()
```

```
  for(i in 1:length(paths)){
    format_one_table = get_one_table(paths[i])
    data_final = rbind(data_final,format_one_table)
  }
  data_final
}
########## step 5: transform variables and generate new variables to analyze #
options(stringsAsFactors = FALSE)
data = Get_all_table(paths)
## transform some columns into approporiate data type
data[,c("place","ag")] = sapply(data[,c("place","ag")],as.numeric)
data[,c("name","hometown")] = sapply(data[,c("name","hometown")],as.character)
data$year = as.numeric(paste(data$year))
## some years don't have net_time or gun_time,using time variable as
## substitute.Create a new varaible:race_time.For years having gun_time:
## race_time=gun_time. For years without gun_time: race_time = time.
race_time=c()
race_time = sapply(1:nrow(data),function(i){
  if(is.na(data[i,"gun"])) race_time[i]= data[i,"time"]
  else race_time[i] = data[i,"gun"]})
data = cbind(data,race_time)
## convert time into numeric value using "convert_time" function
data[,c("time","gun","net","race_time")] =
  sapply(data[,c("time","gun","net","race_time")],convert_time)


################## functions to call in each step ###########################


############## 1.process_file ###############
```

```
##get rid of "#","*" ,lines that are all blank and footnote.Then save
##as a new temporary file in order to generate dataframe for
## the origin file.
process_file = function(filename){
  ## read the file in
  file = readLines(filename)
  ## get rid of footnote
  if(length(grep("^[#\\*] U[A-z].|^[0-9]/.",file))!=0){
    file = file[-c(grep("^[#\\*] U[A-z].|^[0-9]/.",file))]
    ## get rid of # and * in data
    file = gsub("[#*]"," ",file)
    ## get rid of blank lines
    blanks = grep("^[[:blank:]]*$",file)
    file = file[-c(blanks)]
  }
  else {
    file = gsub("[#*]"," ",file)
    # get rid of blank lines
    blanks = grep("^[[:blank:]]*$",file)
    file = file[-c(blanks)]
  }
  writeLines(file,"file_adj")
}
##############
############# 2. line_skip ######################
#####find the "==" row in each file
line_skip = function(file){
  grep("==",file)
```

```
}
#############
################ 3.width_get ########################
######### generate width for read.fwf function
width_get = function(data,index){
  column_split = data[index]
  split = strsplit(column_split," {1,}")
  w = sapply(1:length(split[[1]]),function(x) nchar(split[[1]][x]))
  w = w+1
  w
}
###################
############### 4. match_header ########################
## match each dataframe's header with colnames. Get the column names that
## exist in the table and column names that don't exist in the table.
match_header = function(headRow,col_name){
  no_name=character()
  have_name=character()
  for (i in 1: length(col_name)){
    match = regexpr(col_name[i],headRow)[[1]]
    if(match!=-1) {
      have_name = c(have_name,col_name[i])
    }
    else no_name = c(no_name,col_name[i])
  }

  list(have_name,no_name)
}
```

```
####################

############ 5.data_sub ##############

## get the names from each dataframe's column names(these names represent that

## these columns are in the original dataframe already)

data_sub = function(head_split,data){

  save_name = character()

  head_yes = head_split

  var_name = colnames(data)

  for(i in 1:length(var_name)){

    match = regexpr(substring(var_name[i],1,2),head_yes)

    if(length(unique(match))==2) save_name = c(save_name,var_name[i])

  }

  save_name

}

#################

############# 6. bind_data #############

###function to bind specified vector with existed dataframe

bind_data = function(head_no,men_sub){

  non_matrix = matrix(NA,nrow=nrow(men_sub),ncol=length(head_no))

  non_matrix = as.data.frame(non_matrix)

  colnames(non_matrix) = head_no

  cbind(men_sub,non_matrix)

}

#############

########## 7.year_sex ##############

#### function to add year and sex

year_sex = function(data,filename){

  if(grepl("^men",filename)) data = cbind(data,sex=rep("male",nrow(data)))
```

```
  else data = cbind(data,sex=rep("female",nrow(data)))
  years = gsub("[A-z]+[0-9][0-9][A-z]+_","",filename)
  data = cbind(data,year=rep(years,nrow(data)))
}

##########
########## 8.split_home_nettime #############
## split hometown and net_time in year2006
split_home_nettime = function(data){
  vector = as.character(data$hometownnettim)
  result = strsplit( gsub("([0-9]*:*[0-9]*:[0-9]*)","~\\1~",vector), "~" )
  net = c()
  hometown =c()
  for(i in 1:length(result)){
    hometown[i] = result[[i]][1]
    net[i] = result[[i]][2]
  }
  cbind(data[,-5],hometown,net)
}
##################
######## 8. convert_time ################
## function to convert time into minute
convert_time = function(time_input){
  time_input = as.character(time_input)
  con_time = rep(0,length(time_input))
  for(i in 1: length(time_input)){
    split_time = as.numeric(strsplit(time_input[i],':')[[1]])
    if(length(split_time) >2){
```

```
    con_time[i] = split_time[1]*60 + split_time[2] + split_time[3]/60
  }
  else{
    con_time[i] = split_time[1] + split_time[2]/60
  }
 }
 con_time
}
```

Appendix 2: creating derived variables and creating the summaries I included

```
############### This part is for exploratory data analysis ################
######### Question: Explore age distribution: histogram,median,mean,etc. ############
library(survival)
library(doBy)
mean_age_sex = summaryBy(ag ~ sex, data = data, FUN = list(mean,median),na.rm=TRUE)
mean_age_sex
## boxplot to show the result
## By sex
ggplot(data[-c(which(is.na(data$ag)))],aes(x=sex,y=ag,fill=sex))+geom_boxplot()+stat_summary(fun.y=mean,
geom="point", shape=5, size=4)+ylab("Age")+xlab("Gender")+ggtitle("Age distribution by Gender")
### By year
ggplot(data,aes(x=year,y=ag,fill=year))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=5,
size=4)+facet_grid(.~year,scales="free", space="free")+theme(axis.ticks = element_blank(), axis.text.x =
element_blank())+ylab("age")+ggtitle("Age distribution by Year")
## age ranges are quite similiar in different males and females.
## average age by sex and year
mean_year = summaryBy(ag ~ year+sex, data = data, FUN = list(mean),na.rm=TRUE)
mean_year
mean_year[which(mean_year$sex=="female"),"ag.mean"]
mean_year[which(mean_year$sex=="male"),"ag.mean"]
## boxplot to visuliaze the result
ggplot(data,aes(x=sex,y=ag,fill=sex))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=5,
size=4)+facet_grid(.~year,scales="free", space="free")
## ANSWER: The average ages of male runners are larger than female runners' in all years.
######## Plot the distributions of age  ################
library(ggplot2)
## overall distribution
ggplot(data=
data,aes(x=ag))+geom_histogram(aes(y=..density..),binwidth=0.5,color="black",fill="white")+geom_vline(aes(xinte
rcept=mean(ag, na.rm=T)),color="red", linetype="dashed", size=1)+geom_vline(aes(xintercept=median(ag,
na.rm=T)),color="blue", linetype="dashed", size=1) +xlab("age")+ggtitle("Overall Age Distribution ")
## overall distribution: right skewed.
## distribution of age by sex (histrogram)
ggplot(data=data,aes(x=ag,fill=sex))+geom_histogram(binwidth=0.7,alpha=0.5,position="identity")+ggtitle("Age
distribution by sex")
```

```
#### male runners age distributions are right skewed.
## distribution  of age by year (histogram)
ggplot(data, aes(x=ag)) + geom_histogram(binwidth=.6, colour="blue", fill="white") + facet_grid(.~ year,scales="free", space="free")
#### each year's distributions are quite similiar. But there are more runners in the recent years.
## distribution in each year for male and female
ggplot(data, aes(x=ag)) + geom_histogram(binwidth=.6, colour="black", fill="white") + facet_grid(sex~year,scales="free", space="free")
## It seems that runners are younger in each year for female. Because ##the peak is left skewed. The distributions are all slightly right-skewed.
library(plyr)
######### Question:   Explore race time  difference #######
library(reshape2)
#### Any difference between male and female gun_time(net_time)
summaryBy(race_time~ sex, data = data, FUN = list(mean,median),na.rm=TRUE)
ggplot(data,aes(x=sex,y=race_time,fill=sex))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=7, size=4)+ylab("Race Time")+
  ggtitle("Box plot of Race Time by Sex")
## boxplot of race time
ggplot(data,aes(x=year,y=race_time,fill=year))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=5, size=4)+facet_grid(.~year,scales="free", space="free")+theme(axis.ticks = element_blank(), axis.text.x = element_blank())+ylab("Race Time")+ggtitle("Race time distribution by Year")
## histrogram of race time
ggplot(data=data,aes(x=race_time,fill=sex))+geom_histogram(binwidth=0.7,alpha=0.5,position="identity")+ggtitle( "Race Time distribution by sex")
ggplot(data=data,aes(x=race_time))+geom_histogram(binwidth=0.7,alpha=0.5,position="identity")+ggtitle("Overall Race Time Distribution")+geom_vline(aes(xintercept=mean(race_time, na.rm=T)),color="red", linetype="dashed", size=1)+geom_vline(aes(xintercept=median(race_time, na.rm=T)),color="blue", linetype="dashed", size=1) +xlab("Race Time")+ggtitle("Overall Race Time Distribution ")
#### Mean and median of race time by year
time_year = summaryBy(race_time~ year, data = data, FUN = list(mean,median),na.rm=TRUE)
time_year
## time trend of mean of race time in each year
ggplot(time_year,aes(x=year,y=race_time.mean))+geom_line(linetype="dashed",color="blue",size=1)+ggtitle("Mea n of race time in each year")+ylab("Race Time")+geom_smooth()
#### any difference in race_time for different sex
time_sex = summaryBy(race_time~ sex, data = data, FUN = list(mean,median),na.rm=TRUE)
## female runners' time is longer than male runners.
#### any difference in gun_time for different sex and year
mean_time_sex = summaryBy(race_time~ sex+year, data = data, FUN = list(mean),na.rm=TRUE)
mean_time_women = mean_time_sex[1:12,]
mean_time_men = mean_time_sex[13:24,]
### time trend of mean race time in each year
ggplot()+geom_line(data=mean_time_men,aes(x=year,y=race_time.mean,color="male"))+geom_line(data =mean_time_women,aes(x=year,y=race_time.mean,color="female"))+ggtitle("Average race time in each year")+ylab("average race time")+scale_color_discrete(name="Sex")
## time trend of mean race time by sex in each year
ggplot(data,aes(x=year,y=race_time,fill=year))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=5, size=4)+facet_grid(.~year,scales="free", space="free")+theme(axis.ticks = element_blank(), axis.text.x = element_blank())+ggtitle("boxplot of race_time")
## median gun_times along with year don't change a lot for both male and female runners.
### Male runners spend less time to finish the game. The average gun_time increases in each year.
########### Question: Is age and race_time positively related?
ggplot(data,aes(x=ag,y=race_time,fill=sex))+geom_point()+facet_grid(.~sex,scales="free", space="free")+geom_smooth(method="lm", fill=NA)+xlab("age")+ylab("race time")+ggtitle("Relation between race time and age")
```

```
## For both male and female runners, age has positive relation with gun_time, which means age has negative relation with speed.runners slow as they age
race_age = lm(race_time~ag,data=data)
a = summary(race_age)
## cut age into several group and explore the relation between age and race time in each group
age_group = cut(data$ag, breaks = c(seq(10, 90, 10)))
table(age_group)
## combine with original data
data = cbind(data,age_group)
DATA = data[which(!is.na(data$age_group)),]
## boxplot of race time with each year group
ggplot(DATA,aes(x=age_group,y=race_time))+geom_boxplot()+stat_summary(fun.y=mean, geom="point", shape=5, size=4,na.rm=TRUE)+
  ylab("Race Time")+ggtitle("Race time distribution by age group")+xlab("Age Group")+
  theme(axis.ticks = element_blank(), axis.text.x = element_blank())+theme(axis.ticks = element_blank(), axis.text.x = element_blank())
######### Question : Who are the top local finishers in first two years and last two years?
index_local = grep(". [A-Z][A-Z]",data$hometown)
data_sub = data[index_local,]
data_sub = data_sub[order(data_sub$year,data_sub$race_time),]
Year = paste(1999:2010)
names = c()
for (i in 1:12){
  sub = subset(data_sub,data_sub$year==Year[i])
  names = append(names,paste(sub$year[1],sub$name[1],sub$hometown[1],sub$race_time[1],sub$sex[1]))
}
## save each year's domestic chamption in a data frame
names
# Question: Who are the top international player in different years?

data_int = data[-index_local,]
name_int = c()
for (i in 1:12){
  sub = subset(data_int,data_int$year==Year[i])
  name_int = append(name_int,paste(sub$year[1],sub$name[1],sub$hometown[1],sub$race_time[1]))
}
name_int
######### Question:Who are the winners of women group in each year?
data_int_women = subset(data_int,data_int$sex=="female")
name_women_int = c()
for (i in 1:12){
  sub = subset(data_int_women,data_int_women$year==Year[i])
  name_women_int = append(name_women_int,paste(sub$year[1],sub$name[1],sub$hometown[1],sub$race_time[1]))
}
name_women_int
############## Question: scatter plot of race_time vs. age
smoothScatter(y = data$race_time, x = data$ag,
          ylim = c(40, 165), xlim = c(15, 85),
          xlab = "Age (years)", ylab = "Race Time (minutes)",bandwidth=c(2,1)/3,main="Scatter plot: Age vs. Race Time")
######## QUESTION 11: fitting models for average performance
race_age = lm(race_time~ag,data=data)
summary(race_age)
##### QUESTION How many entrants are there over the 14 years?
## write a function to clean the name and hometown variable
```

```
eli_blanks = function(vector) {
  nameclean = gsub("^[[:blank:]]+", "", vector)
  nameclean = gsub("[[:blank:]]+$", "", nameclean)
  nameclean = gsub("[[:blank:]]+", " ", nameclean)
  nameclean = tolower(nameclean)
  nameclean = gsub("[,.]", "", nameclean)
}
name_clean = eli_blanks(data$name)
name_clean
hometown_clean = eli_blanks(data$hometown)
## combind the clean version of name and hometown
data=data_back
data = cbind(data,name_clean)
data = cbind(data,hometown_clean)
## there are 113190 entrants in total
length(name_clean)
##### How many unique names are there among these entrants?
length(unique(name_clean))
## there are 75382 unique names
##### How many names appear twice, 3 times, 4 times, etc. and what #name occurs most often?
table(table(name_clean))
#We see that over 13000 names appear 2 times throughout the 12 races. #One name appears 26 times, and we know
this name must correspond to #at least 2 people because we have only 12 years of race results.
################### create the new ID ########################
## use the clean version of name to create the unique ID
## generate a new variable:birth(represent birth year for each obs)
data$birth = data$year - data$ag
## generate the unique ID combing birth and name
data$id_birth = paste(data$name_clean,data$birth,sep = "_")
## how many unique IDs
length(unique(data$id_birth))
table(table(data$id_birth))
#Since our goal is to study how an athlete's time changes with age, #let's focus on those IDs that appear in at least 10
races.
races = tapply(data$year, data$id_birth, length)
races10 = names(races)[which(races >= 10)]
runner10 = data[ data$id_birth %in% races10, ]
runner10 = runner10[order(runner10$id_birth, runner10$year), ]
runner10
### how many unique ids for people who attends the game at least 10 times ?
length(unique(runner10$id_birth))
## what are these players' names and their year of birth?
unique(runner10$id_birth)
## explore people who attend 12 times
races12 = names(races)[which(races == 12)]
runner12 = data[ data$id_birth %in% races12, ]
runner12 = runner12[order(runner12$id_birth, runner12$year), ]
runner12

ggplot(aes(x=year,y=race_time,group=id_birth,color=id_birth),data=runner12)+ geom_step(direction =
"hv")+geom_smooth()+
  facet_wrap(~id_birth,nrow=4)+ggtitle("Time Trend for players")
```