Cherry Blossom Ten Mile Run & Walk

Examining the Impact of Age on Physical Performance

The Raw Data

data is pre-processed from the web in \02_data.r

```
load_results <- function(class, year) {

   folder <- ifelse(class == "men", "men_txt", "women_txt")
   file <- file.path(data.dir, folder, paste(year, ".txt", sep = ""))

   return(readLines(file))
}
```

Race Results Preprocessing

```
els <- load_results("men", 2006)

eqIndex <- grep("^===", els)
eqIndex
```

```
[1] 8
```

```
spacerRow <- els[eqIndex]
headerRow <- els[eqIndex - 1]

body <- els[ -(1:eqIndex) ]

headerRow <- tolower(headerRow)

timeStart <- regexpr("net", headerRow)
time <- substr(body, start = timeStart, stop = timeStart + 1)

ageStart <- regexpr("ag", headerRow)
```

```
age <- substr(body, start = ageStart, stop = ageStart + 1)
head(age)
```

```
[1] "27" "29" "23" "28" "28" "29"
```

```
summary(as.numeric(age))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   2.00    3.00   34.00   28.53   46.00   82.00       2
```

```
blankLocs <- gregexpr(" ", spacerRow)
blankLocs
```

```
[[1]]
 [1]  6 15 22 45 48 64 72 80 81 87 89
attr(,"match.length")
 [1] 1 1 1 1 1 1 1 1 1 1 1
attr(,"index.type")
[1] "chars"
attr(,"useBytes")
[1] TRUE
```

```
searchLocs <- c(0, blankLocs[[1]])

Values <- mapply(substr, list(body),
                 start = searchLocs[ -length(searchLocs)] + 1,
                 stop = searchLocs[ -1 ] - 1)

findColLocs <- function(spacerRow) {
   spaceLocs <- gregexpr(" ", spacerRow)[[1]]
   rowLength <- nchar(spacerRow)

   if(substring(spacerRow, rowLength, rowLength + 1) != " ")
      return(c(0, spaceLocs, rowLength + 1))
   else
      return(c(0, spaceLocs))
}

findColLocs(spacerRow)
```

```
 [1]  0  6 15 22 45 48 64 72 80 81 87 89
```

```
name <- "home"
colnames <- c("name", "home", "ag", "gun", "net", "time")

colIndex <- which(colnames == name)
```

```
startPos <- regexpr(name, headerRow)[[1]]

#
# can we modify select cols to short circuit on matching w/o spaces?
#######


selectCols =
   function(colNames, headerRow, searchLocs)
   {
      sapply(colNames,
             function(name, headerRow, searchLocs)
            {
               startPos <- regexpr(name, headerRow)[[1]]

               if(startPos == -1)
                  return( c(NA, NA) )

               index <- sum(startPos >= searchLocs)

               c(searchLocs[index] + 1, searchLocs[index + 1])
            },
      headerRow = headerRow, searchLocs = searchLocs)
   }

searchLocs <- findColLocs(spacerRow)
loc <- selectCols("home", headerRow, searchLocs)
vars <- mapply(substr, list(body), start = loc[1,], stop = loc[2, ])

summary(as.numeric(vars))
```

```
Warning in summary(as.numeric(vars)): NAs introduced by coercion
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
     NA      NA      NA     NaN      NA      NA    5237
```

```
shortColNames <- c("name", "home", "ag", "gun", "net", "time")

locCols <- selectCols(shortColNames, headerRow, searchLocs)
Values <- mapply(substr, list(body), start = locCols[1, ],
                 stop = locCols[2, ])

class(Values)
```

```
[1] "matrix" "array"
```

```
colnames(Values) = shortColNames

head(Values)
```

```
     name                      home               ag    gun         net
[1,] "Gilbert Okari          " " "Kenya           " " "27 " " "  47:25#" "  47:24 "
[2,] "Samuel Ndereba         " " "Kenya           " " "29 " " "  47:35#" "  47:34 "
[3,] "Rueben Kibet Chebii    " " "Kenya           " " "23 " " "  47:39#" "  47:38 "
[4,] "Kazuo Ietani           " " "Japan           " " "28 " " "  47:39#" "  47:39 "
[5,] "Wilson Komen           " " "Kenya           " " "28 " " "  47:58#" "  47:58 "
[6,] "Matt Downin            " " "United States   " " "29 " " "  48:43#" "  48:42 "
     time
[1,] NA
[2,] NA
[3,] NA
[4,] NA
[5,] NA
[6,] NA
```

```
tail(Values)[, 1:3]
```

```
        name                      home               ag
[5232,] " Ishong Nkong          " " " Herndon        " " 29"
[5233,] "Ted Whichard           " " "Roanoke         " " "53 "
[5234,] "Doug Whichard          " " "Blacksburg      " " "60 "
[5235,] "Daniel Grasso          " " "Lanham          " " "50 "
[5236,] "deline    "              ""                   ""
[5237,] "p guideline"             ""                   ""
```

```
extractVariables =
   function(file, varNames = c("name", "home", "ag", "gun", "net", "time"))
   {
      # Find the index of the row with =s
      eqIndex <- grep("^===", file)
      spacerRow <- file[eqIndex]
      headerRow <- tolower(file[ eqIndex - 1 ])
      body <- file[ -(1 : eqIndex) ]

      blank <- grep("^[[:blank:]]*$", body)
      footnote <- grep("^[^\\s]*[\\*]*[#]", body)
      ignore <- union(blank, footnote)

      if(length(ignore))
         body <- body[-ignore]

      # Obtain the starting and ending positions of variables
```

```
      searchLocs <- findColLocs(spacerRow)
      locCols <- selectCols(varNames, headerRow, searchLocs)

      Values <- mapply(substr, list(body), start = locCols[1, ],
                       stop = locCols[2, ])
      colnames(Values) <- varNames

      invisible(Values)
   }

load_data <- function(class) {
   folder <- ifelse(class == "men", "men_txt", "women_txt")
   filenames <- file.path(data.dir, paste(folder, "/", 1999:2012, ".txt", sep = ""))
   files <- lapply(filenames, readLines)
   names(files) <- 1999:2012

   mat <- lapply(files, extractVariables)

   return(mat)
}
```

Data Cleaning and Reformatting Variables

```
menResMat <- load_data("men")

length(menResMat)
```

```
[1] 14
```

```
sapply(menResMat, nrow)
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
3190 3004 3546 3723 3939 4156 4325 5230 5264 5903 6641 6899 7010 7193
```

```
womenResMat <- load_data("women")

length(womenResMat)
```

```
[1] 14
```

```
sapply(womenResMat, nrow)
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
3190 3004 3546 3723 3939 4156 4325 5230 5264 5903 6641 6899 7010 7193
```

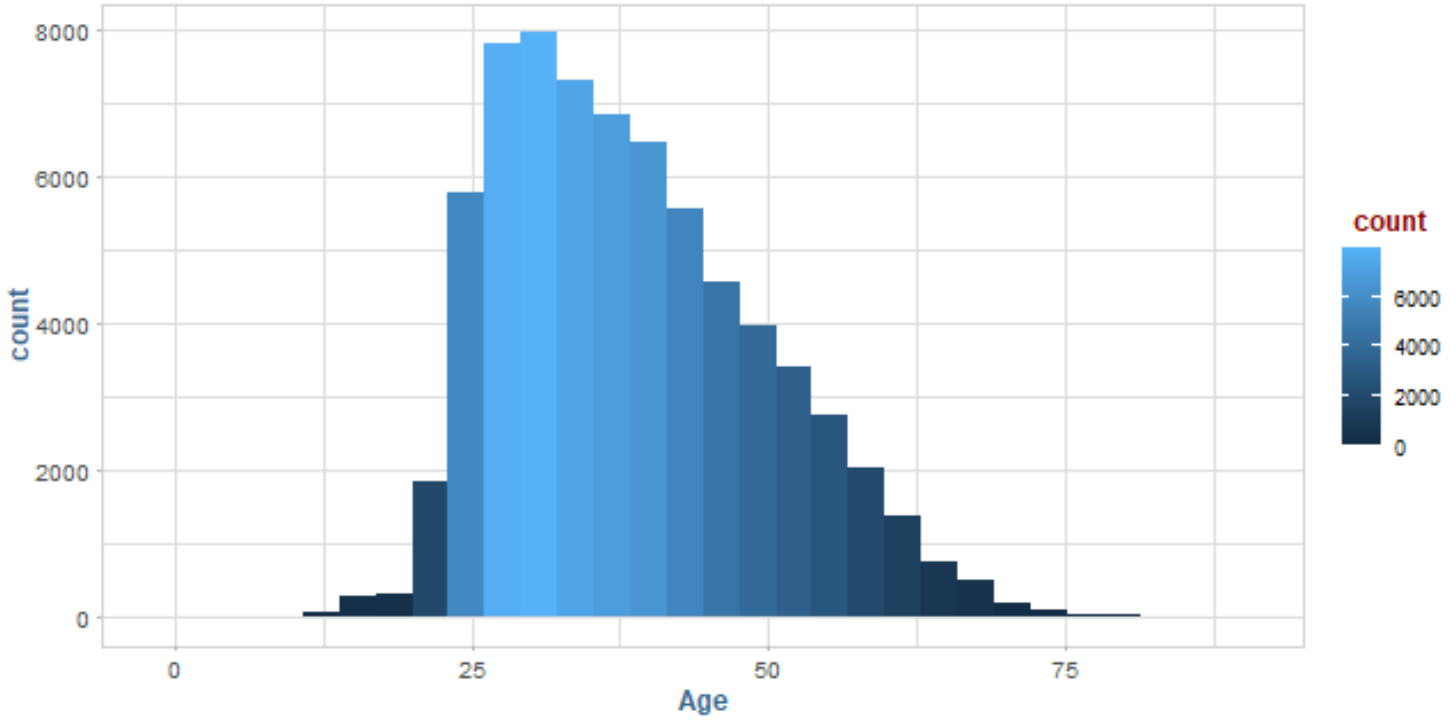
Age Validation

```
age <- as.numeric(menResMat[['2012']][, 'ag'])

tail(age)
```

```
[1] 41 39 56 35 NA 48
```

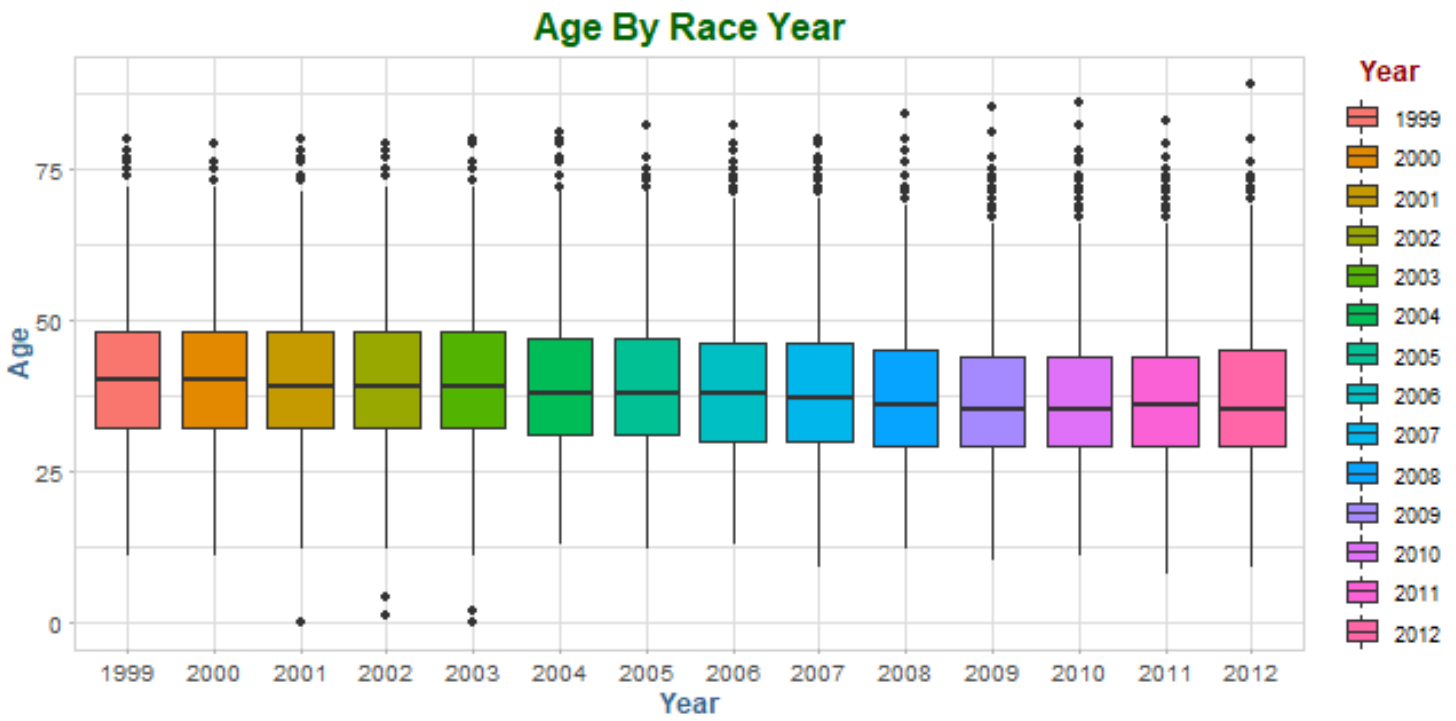
```
age <- sapply(menResMat,
              function(x) as.numeric(x[, "ag"]))

age_values <- plyr::ldply(age, data.frame)
colnames(age_values) <- c("Year", "Age")

ggplot(age_values, aes(Age)) +
   geom_histogram(aes(fill = ..count..), bins = 30)
```

```
Warning: Removed 29 rows containing non-finite values (stat_bin).
```




```
ggplot(age_values, aes(Year, Age)) +
   geom_boxplot(aes(fill = Year)) +
   labs(title = "Age By Race Year")
```

```
Warning: Removed 29 rows containing non-finite values (stat_boxplot).
```




```
sapply(age, function(x) sum(is.na(x)))
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   1    0    0    2    1    0   11    1    4    1    1    5    1    1
```

```
file2001 <- load_results("men", 2001)
age2001 <- as.numeric(extractVariables(file2001)[, "ag"])

badAgeIndex <- which(is.na(age2001)) + 5

file2001[badAgeIndex]
```

```
character(0)
```

```
badAgeIndex
```

```
numeric(0)
```

```
blanks <- grep("^[[:blank:]]*$", file2001)
blanks
```

```
 [1]    1    2    3 1756 1757 1758 1759 1760 1761 1762 1763 1814 1815 1816 1817
[16] 1818 1819 1820 1821 1872 1873 1874 1875 1876 1877 1878 1879 1930 1931 1932
[31] 1933 1934 1935 1936 1937 2538 2539 2540 2541 2542 2543 2544 2545 2546 2897
[46] 2898 2899 2900 2901 2902 2903 2904 2955 2956 2957 3008 3009 3010 3011 3012
[61] 3013 3014 3015
```

```
which(age2001 < 5)
```

```
[1] 1362 2988 3037
```

```
file2001[which(age2001 < 5)]
```

```
[1] " 1357   513 Darin SLADE           34 Falls Church VA    1:21:26 1:22:23"
[2] " 2931  4848 Dave LYTLE            50 Alexandria VA      1:35:27 1:40:10"
[3] " 2972  1648 James COLBY           58 Fairfax VA         1:39:45 1:40:53"
```

```
file <- load_results("men", 2001)
data <- extractVariables(file)
age <- as.numeric(data[, "ag"])

sum(age < 5)
```

```
[1] 3
```

```
which(age == 0)
```

```
[1] 1362 2988 3037
```

```
data[which(age == 0)]
```

```
[1] "Steve PINKOS          " "Jeff LAKE             " "Greg RHODE           "
```

Time Validation

```
file2002 <- load_results("men", 2012)

charTime <- menResMat[["2012"]][, "time"]

head(charTime)
```

```
[1] "  45:15 " "  46:28 " "  47:33 " "  47:34 " "  47:40 " "  47:50 "
```

```
tail(charTime)
```

```
[1] "2:26:47 " "2:27:11 " "2:27:20 " "2:27:30 " "2:28:58 " "2:30:59 "
```

```
timePieces <- strsplit(charTime, ":")

timePieces[[1]]
```

```
[1] "  45" "15 "
```

```
tail(timePieces, 1)
```

```
[[1]]
[1] "2"   "30"  "59 "
```

```
timePieces <- sapply(timePieces, as.numeric)

runTime <- sapply(timePieces,
                  function(x) {
                     if(length(x) == 2) x[1] + x[2]/60
                     else 60 * x[1] + x[2] + x[3]/60
                  })

summary(runTime)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  45.25   77.57   87.47   88.43   97.78  150.98
```

```
convertTime <- function( charTime ) {

   timePieces <- strsplit(charTime, ":")

   timePieces <- sapply(timePieces, as.numeric)

   runTime <- sapply(timePieces,
                  function(x) {
                     if(length(x) == 2) x[1] + x[2]/60
                     else 60 * x[1] + x[2] + x[3]/60
                  })

}
```

Aggregate cleaning into

```
createDF =
   function(Res, year, sex) {
      useTime <- if(!is.na(Res[1, 'net']) )
                     Res[, 'net']
                  else if( !is.na(Res[1, 'gun']) )
                     Res[ , 'gun']
                  else
                     Res[, 'time']

      useTime <- gsub("[#\\*[:blank:]]", "", useTime)

      Res <- Res[ useTime != "", ]

      runTime <- convertTime(useTime[ useTime != "" ])
```

```
      N <- nrow(Res)

      Results <- data.frame( year = rep(year, N),
                             sex = rep(sex, N),
                             name = Res[, 'name'],
                             home = Res[, 'home'],
                             age = as.numeric(Res[, 'ag']),
                             runTime = runTime,
                             stringsAsFactors = F)

      invisible(Results)
   }

years <- 1999:2012
menDF <- mapply(createDF, menResMat, year = years,
                sex = rep("M", 14), SIMPLIFY = F)

warnings()[ c(1:2, 49:50)]
```

```
NULL
```

```
sapply(menDF, function(x) sum(is.na(x$runTime)))
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
file2006 <- load_results("men", 2006)
parsed2006 <- extractVariables(file2006)
time2006 <- parsed2006[, "net"]

womenDF <- mapply(createDF, womenResMat, year = years,
                  sex = rep("F", 14), SIMPLIFY = F)

sapply(womenDF, function(x) sum(is.na(x$runTime)))
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

Consolidate Results

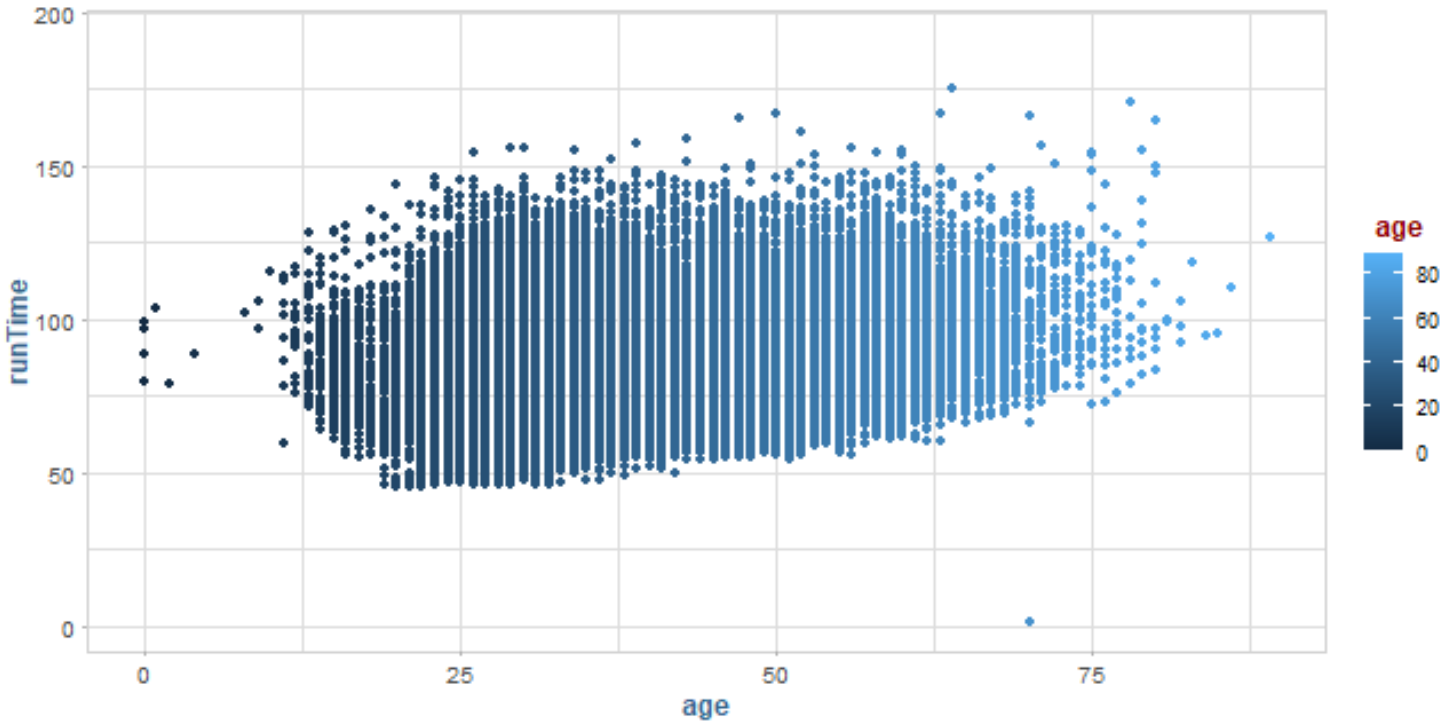
```
cbMen <- do.call(rbind, menDF)
save(cbMen, file = file.path(data.dir, "cbMen.rda"))

cbWomen <- do.call(rbind, womenDF)
```

```
save(cbWomen, file = file.path(data.dir, "cbWomen.rda"))
```

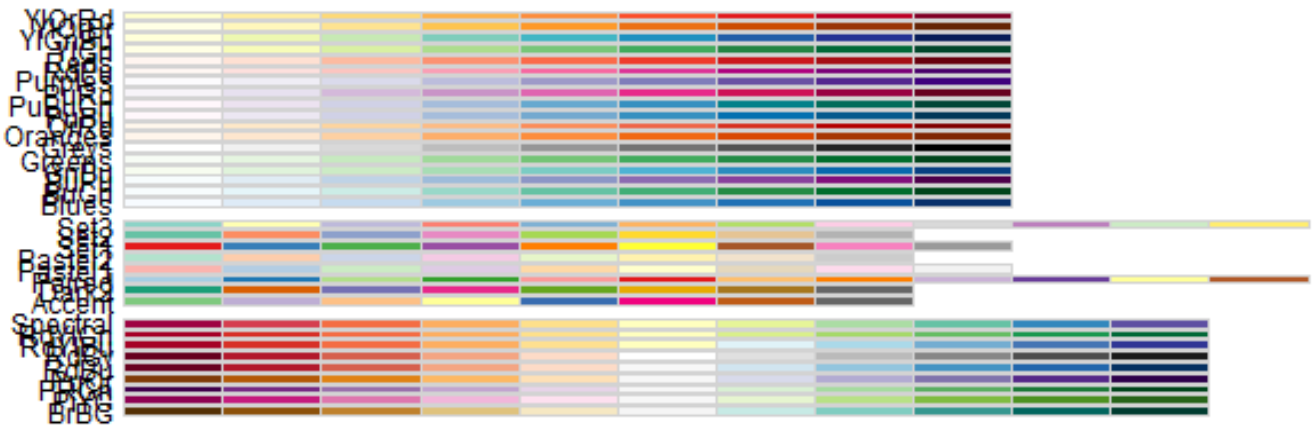
Exploratory Data Analysis

```
ggplot(cbMen, aes(age, runTime)) +
   geom_point(aes(col = age)) +
   labs(main = "Run Times by Age", xlab = "Age (years)", xlab = "Run Time (minutes")
```

```
Warning: Removed 23 rows containing missing values (geom_point).
```

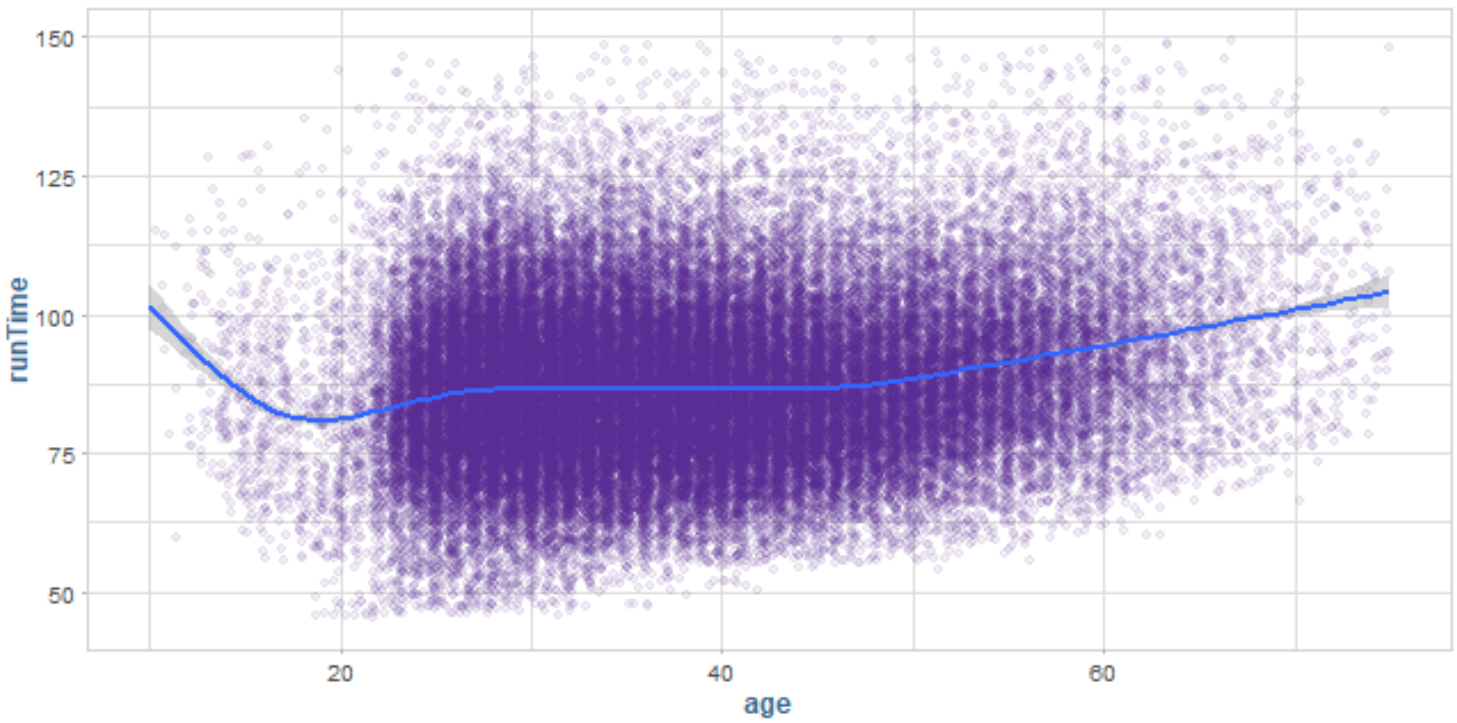



```
display.brewer.all()
```

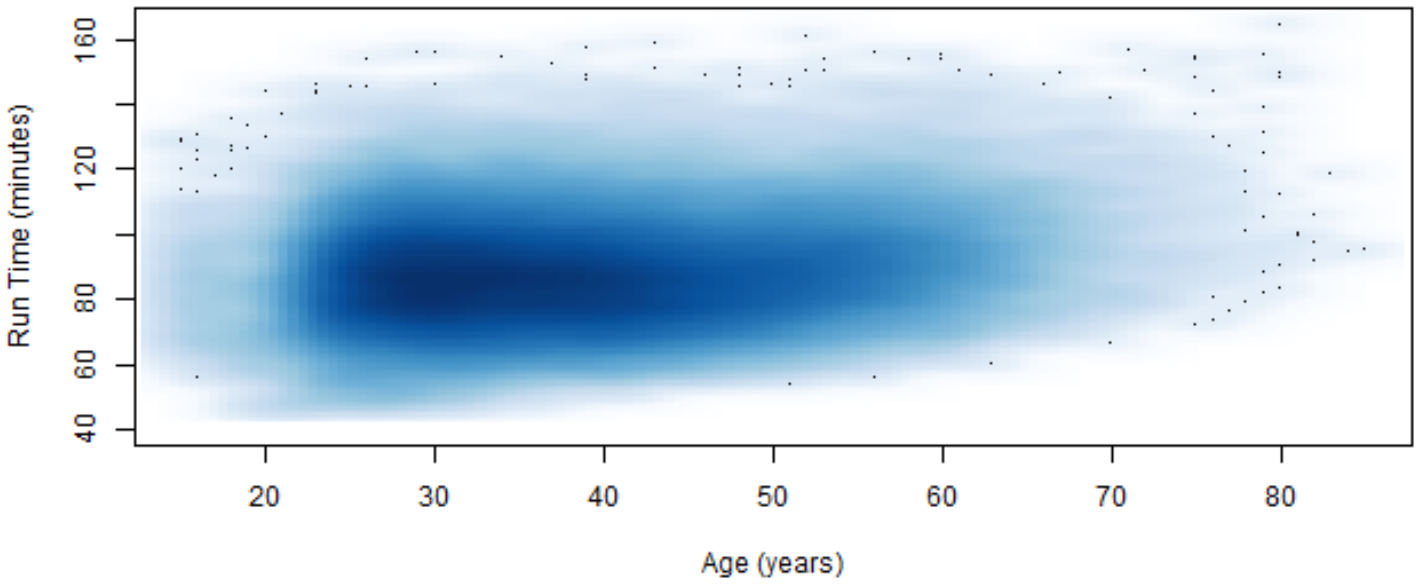


```
Purple8 <- brewer.pal(9, "Purples")[8]

Purple8A <- paste(Purple8, "14", sep = "")

ggplot(cbMen, aes(age, runTime)) +
   geom_jitter(col = Purple8A) +
   geom_smooth() +
   xlim(10, 75) +
   ylim(45, 150)
```

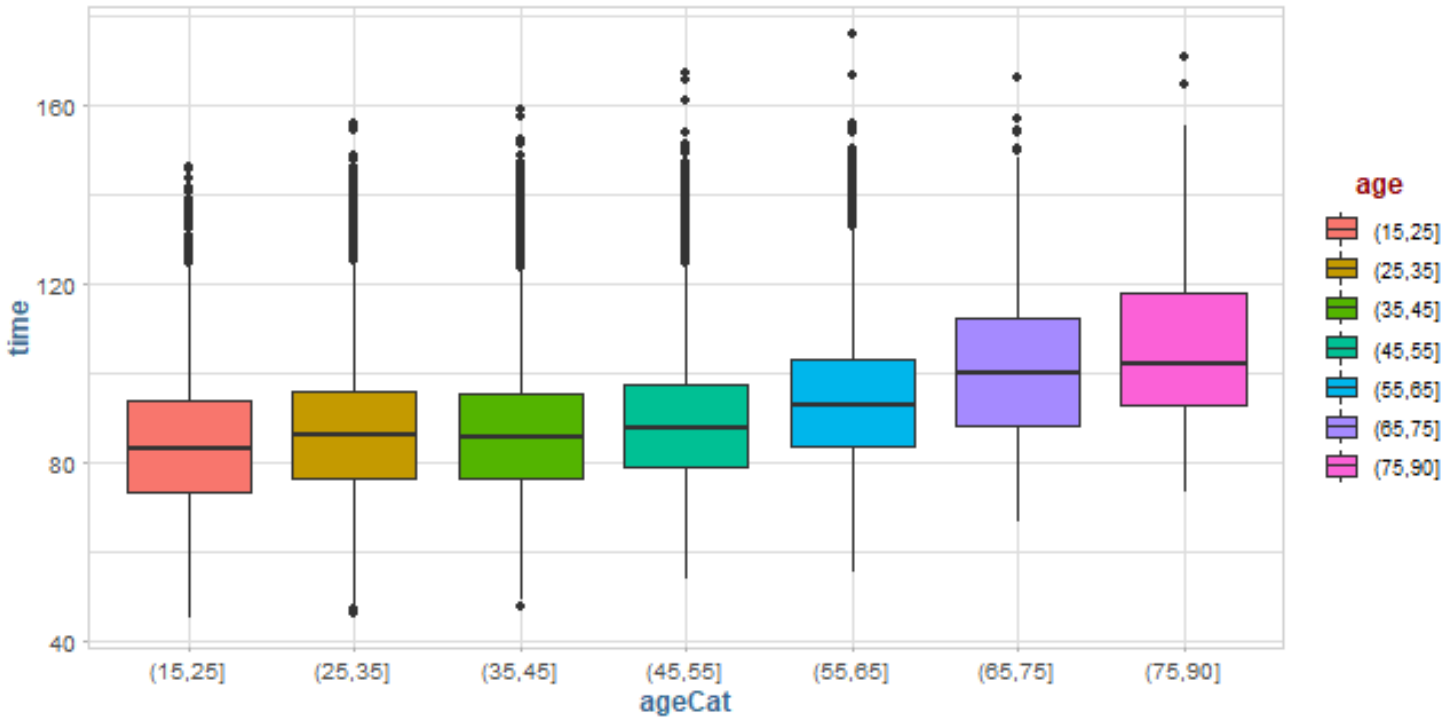
```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
Warning: Removed 130 rows containing non-finite values (stat_smooth).
Warning: Removed 142 rows containing missing values (geom_point).
```




```
smoothScatter(y = cbMen$runTime, x = cbMen$age,
              ylim = c(40, 165), xlim = c(15, 85),
              xlab = "Age (years)", ylab = "Run Time (minutes)")
```



```
cbMenSub <- cbMen[cbMen$runTime > 30 &
                    !is.na(cbMen$age) & cbMen$age > 15,]

ageCat <- cut(cbMenSub$age, breaks = c(seq(15, 75, 10), 90))

ggplot(data.table(age = ageCat, time = cbMenSub$runTime), aes(ageCat, time)) +
   geom_boxplot(aes(fill = age)) +
   labs("Run Times by Age Category")
```

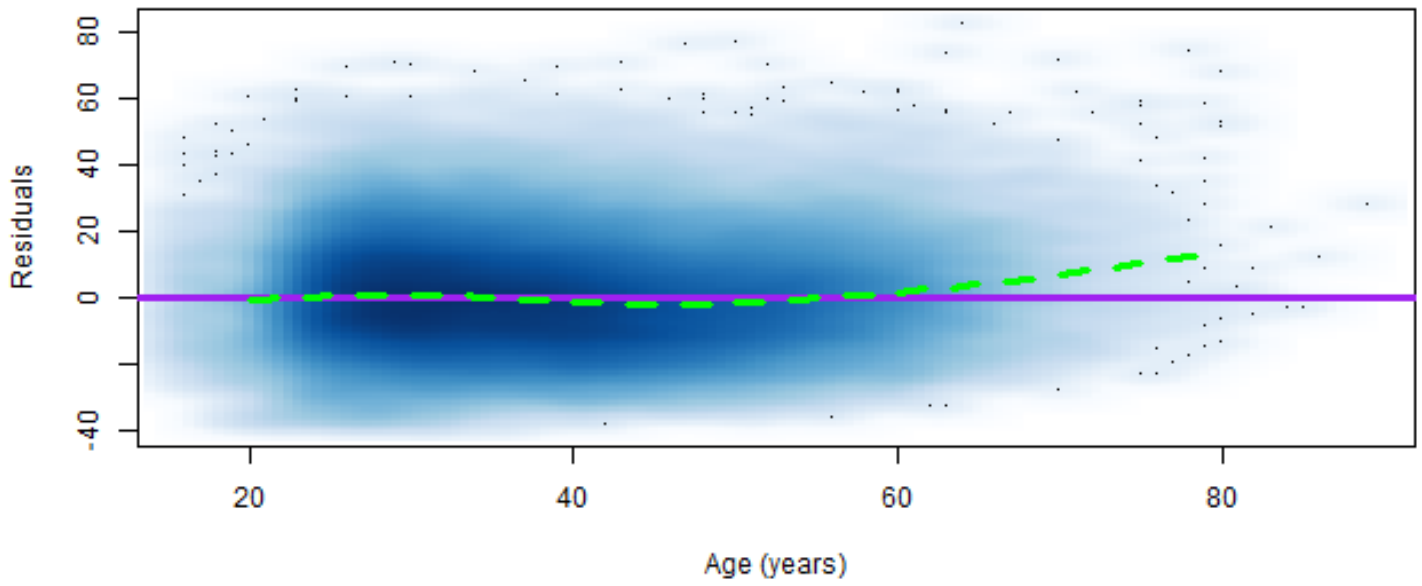



```
lm_age <- lm(runTime ~ age, data = cbMenSub)

summary(lm_age)
```

```
Call:
lm(formula = runTime ~ age, data = cbMenSub)

Residuals:
    Min      1Q  Median      3Q     Max
-39.821 -10.234  -0.972   9.083  82.482

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 78.93366    0.20738  380.62   <2e-16 ***
age          0.22163    0.00516   42.95   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.73 on 69733 degrees of freedom
Multiple R-squared:  0.02577,   Adjusted R-squared:  0.02576
F-statistic:  1845 on 1 and 69733 DF,  p-value: < 2.2e-16
```

```
smoothScatter(x = cbMenSub$age, y = lm_age$residuals,
              xlab = "Age (years)", ylab = "Residuals")
abline(h = 0, col = "purple", lwd = 3)

resid.lo <- loess(resids ~ age,
                  data = data.frame(resids = residuals(lm_age),
                                    age = cbMenSub$age))

age20to80 <- 20:80

resid.lo.pr <- predict(resid.lo, newdata = data.frame(age = age20to80))

lines(x = age20to80, y = resid.lo.pr,
      col = "green", lwd = 3, lty = 2)
```




```
menRes.lo <- loess(runTime ~ age, cbMenSub)

summary(menRes.lo)
```

```
Call:
loess(formula = runTime ~ age, data = cbMenSub)

Number of Observations: 69735
Equivalent Number of Parameters: 5.11
Residual Standard Error: 14.66
Trace of smoother matrix: 5.58  (exact)

Control settings:
  span     :  0.75
  degree   :  2
  family   :  gaussian
  surface  :  interpolate     cell = 0.2
  normalize:  TRUE
 parametric:  FALSE
drop.square:  FALSE
```

```
menRes.lo.pr <- predict(menRes.lo, data.frame(age = age20to80))
```

```
over50 <- pmax(0, cbMenSub$age - 50)

lmOver50 <- lm(runTime ~ age + over50, data = cbMenSub)

summary(lmOver50)
```

```

Call:
lm(formula = runTime ~ age + over50, data = cbMenSub)

Residuals:
    Min      1Q  Median      3Q     Max
-40.118 -10.115  -0.901   9.032  79.083

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 83.002850   0.264631  313.65   <2e-16 ***
age          0.099949   0.007134   14.01   <2e-16 ***
over50       0.573213   0.023314   24.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.67 on 69732 degrees of freedom
Multiple R-squared:  0.03414,   Adjusted R-squared:  0.03412
F-statistic:  1233 on 2 and 69732 DF,  p-value: < 2.2e-16
```

```
decades <- seq(30, 60, by = 10)
overAge <- lapply(decades,
                  function(x) pmax(0, (cbMenSub$age - x)))
names(overAge) <- paste("over", decades, sep = "")
overAge <- as.data.frame(overAge)

tail(overAge)
```

```
      over30 over40 over50 over60
69730     36     26     16      6
69731     11      1      0      0
69732      9      0      0      0
69733     26     16      6      0
69734      5      0      0      0
69735     18      8      0      0
```

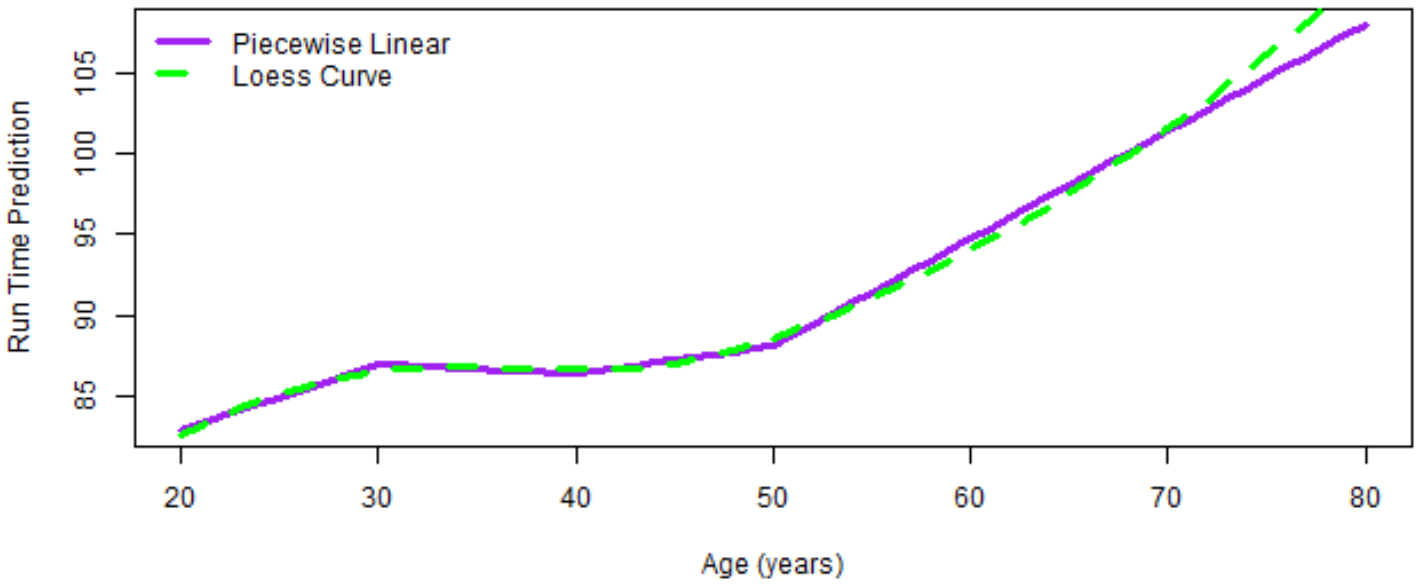
```
lmPiecewise <- lm(runTime ~.,
                  data = cbind(cbMenSub[, c("runTime", "age")],
                               overAge))

summary(lmPiecewise)
```

```
Call:
lm(formula = runTime ~ ., data = cbind(cbMenSub[, c("runTime",
    "age")], overAge))

Residuals:
    Min      1Q  Median      3Q     Max
-40.968 -10.131  -0.905   9.001  78.962

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.784394   0.914733  81.755  < 2e-16 ***
age          0.407548   0.033184  12.282  < 2e-16 ***
over30      -0.466568   0.047720  -9.777  < 2e-16 ***
over40       0.229115   0.040568   5.648 1.63e-08 ***
over50       0.492733   0.052792   9.333  < 2e-16 ***
over60      -0.002991   0.077449  -0.039    0.969
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.66 on 69729 degrees of freedom
Multiple R-squared:  0.03547,   Adjusted R-squared:  0.03541
F-statistic: 512.9 on 5 and 69729 DF,  p-value: < 2.2e-16
```

```
overAge20 <- lapply(decades, function(x) pmax(0, (age20to80 - x)))
names(overAge20) <- paste("over", decades, sep = "")
overAgeDF <- cbind(age = data.frame(age = age20to80), overAge20)

tail(overAgeDF)
```

```
   age over30 over40 over50 over60
56  75     45     35     25     15
57  76     46     36     26     16
58  77     47     37     27     17
59  78     48     38     28     18
60  79     49     39     29     19
61  80     50     40     30     20
```

```
predPiecewise <- predict(lmPiecewise, overAgeDF)

plot(predPiecewise ~ age20to80,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Run Time Prediction")

lines(x = age20to80, y = menRes.lo.pr,
      col = "green", lty = 2, lwd = 3)

legend("topleft", col = c("purple", "green"),
       lty = c(1, 2), lwd = 3,
       legend = c("Piecewise Linear", "Loess Curve"), bty = "n")
```



Cross-Sectional Data and Covariates

```
age1999 <- cbMenSub[ cbMenSub$year == 1999, "age"]
age2012 <- cbMenSub[ cbMenSub$year == 2012, "age"]
```

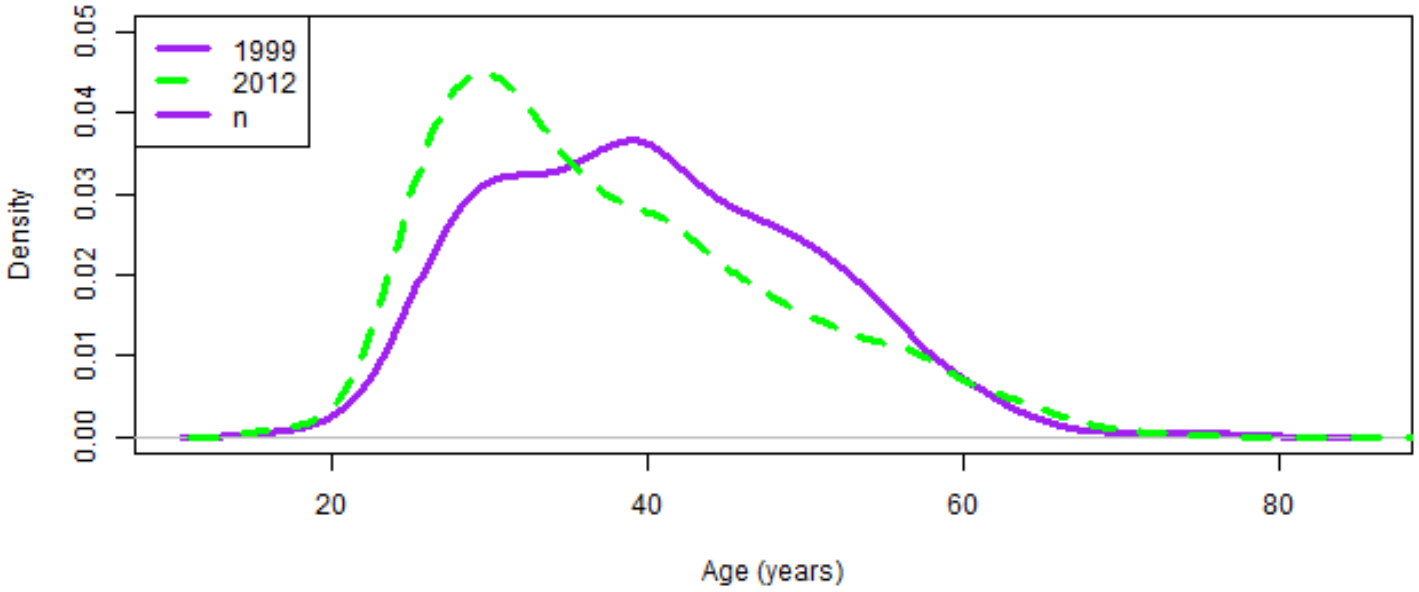
```
summary(age1999)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  16.00   32.00   40.00   40.43   48.00   80.00 
```

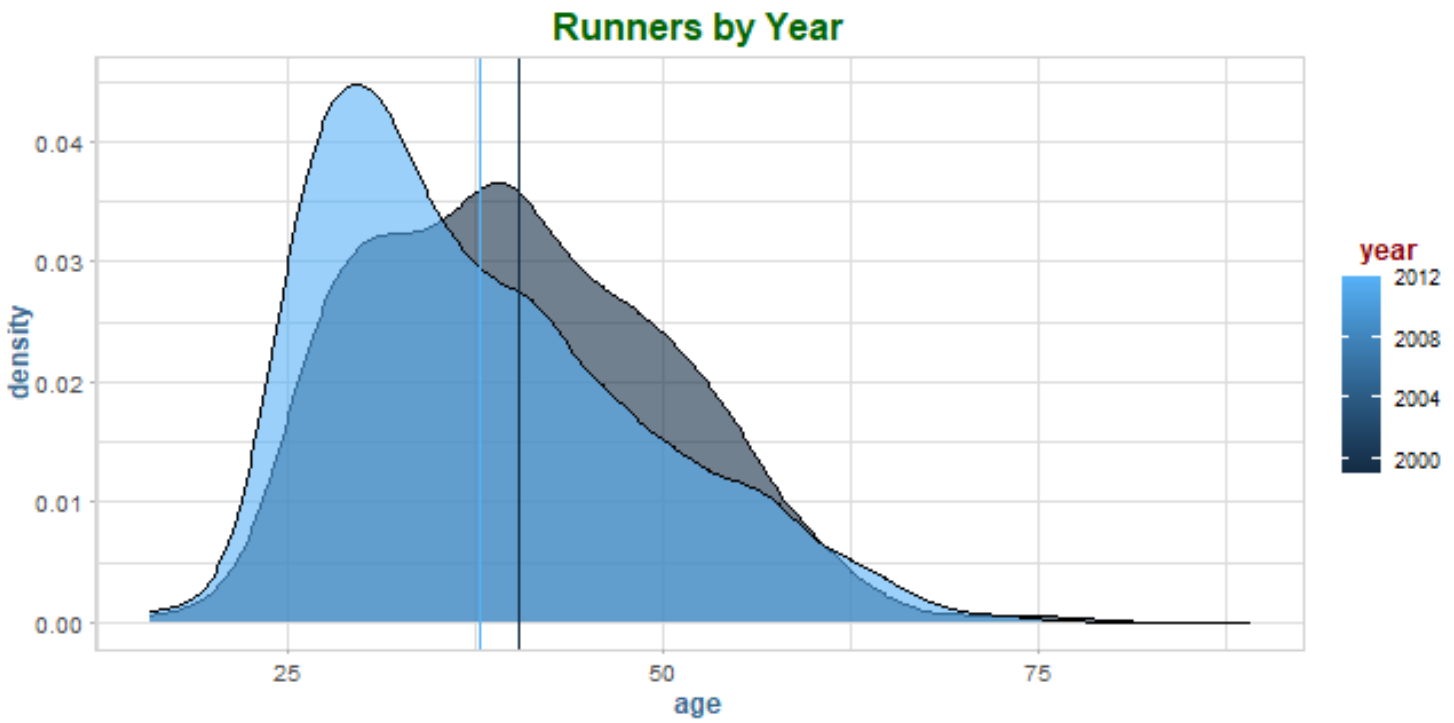
```
summary(age2012)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  16.00   29.00   36.00   37.84   45.00   89.00 
```

```
plot(density(age1999, na.rm = T),
     ylim = c(0, 0.05), col = "purple",
     lwd = 3, xlab = "Age (years)", main = "")
lines(density(age2012, na.rm = T),
      lwd = 3, lty = 2, col = "green")
legend("topleft", col = c("purple", "green"), lty = 1:2, lwd = 3,
       legend = c("1999", "2012", bty = "n"))
```

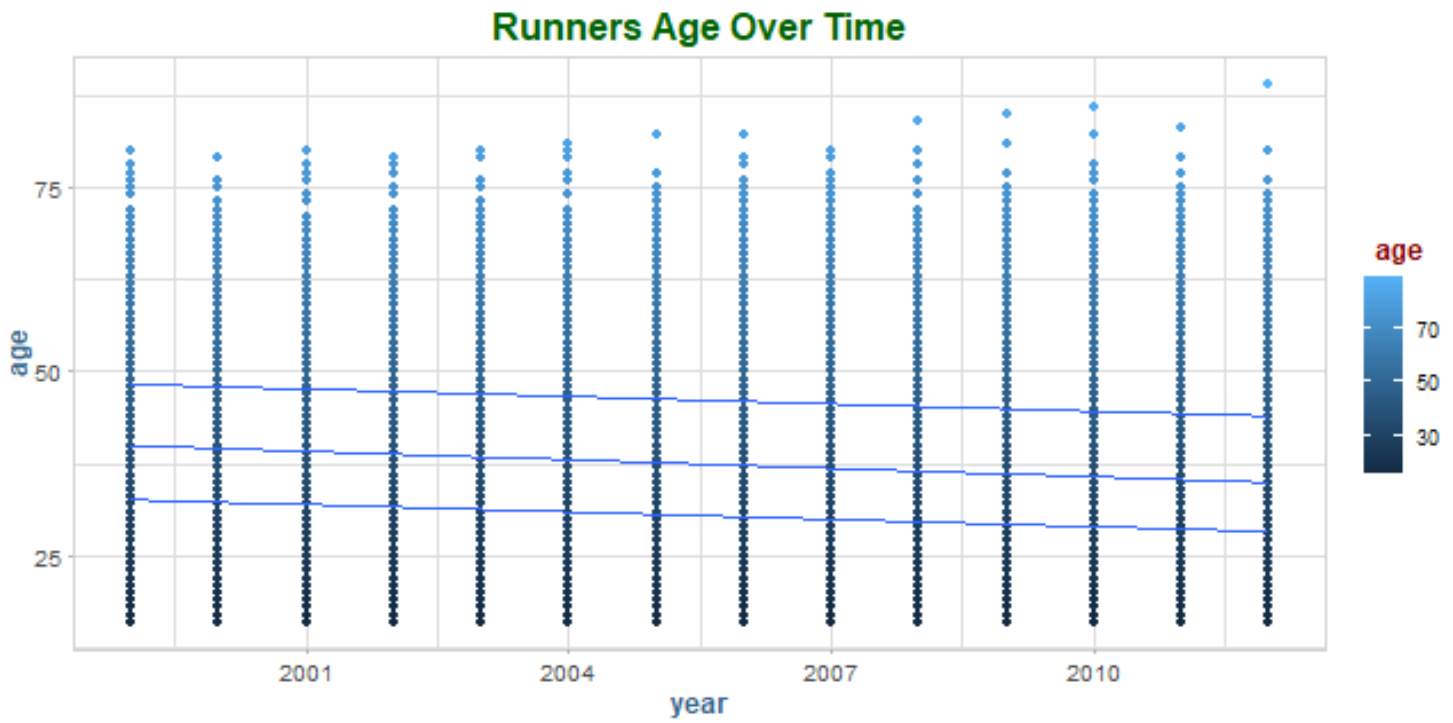



```
by_year <- data.table(cbMenSub)[, .(age, mean = mean(age)), by = list(year)]

ggplot(by_year[year %in% c("1999", "2012")], aes(age, group = year)) +
   geom_density(aes(fill = year), alpha = .6) +
   geom_vline(aes(xintercept = mean, col = year, group = year)) +
   labs(title = "Runners by Year")
```




```
ggplot(by_year, aes(year, age)) +
   geom_point(aes(col = age)) +
   geom_quantile() +
   labs(title = "Runners Age Over Time")
```

```
Smoothing formula not specified. Using: y ~ x
```




```
mR.lo99 <- loess(runTime ~ age, cbMenSub[ cbMenSub$year == 1999,])
mR.lo.pr99 <- predict(mR.lo99, data.frame(age = age20to80))

summary(mR.lo.pr99)
```

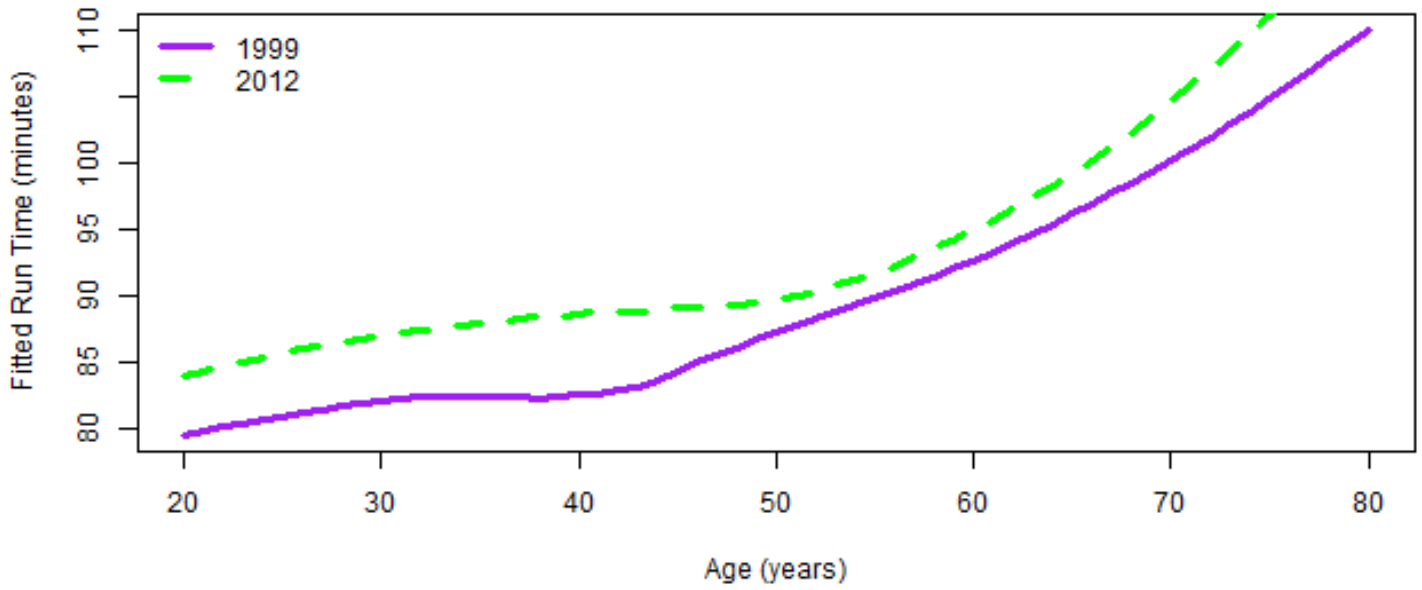
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  79.53   82.42   87.27   89.90   96.15  110.07
```

```
mR.lo12 <- loess(runTime ~ age, cbMenSub[ cbMenSub$year == 2012, ])
mR.lo.pr12 <- predict(mR.lo12, data.frame(age = age20to80))

summary(mR.lo12)
```

```
Call:
loess(formula = runTime ~ age, data = cbMenSub[cbMenSub$year ==
    2012, ])

Number of Observations: 7164
Equivalent Number of Parameters: 5.08
Residual Standard Error: 15.23
Trace of smoother matrix: 5.55  (exact)

Control settings:
  span     :  0.75
  degree   :  2
  family   :  gaussian
```

```
  surface   :  interpolate     cell = 0.2
  normalize:  TRUE
 parametric:  FALSE
drop.square:  FALSE
```

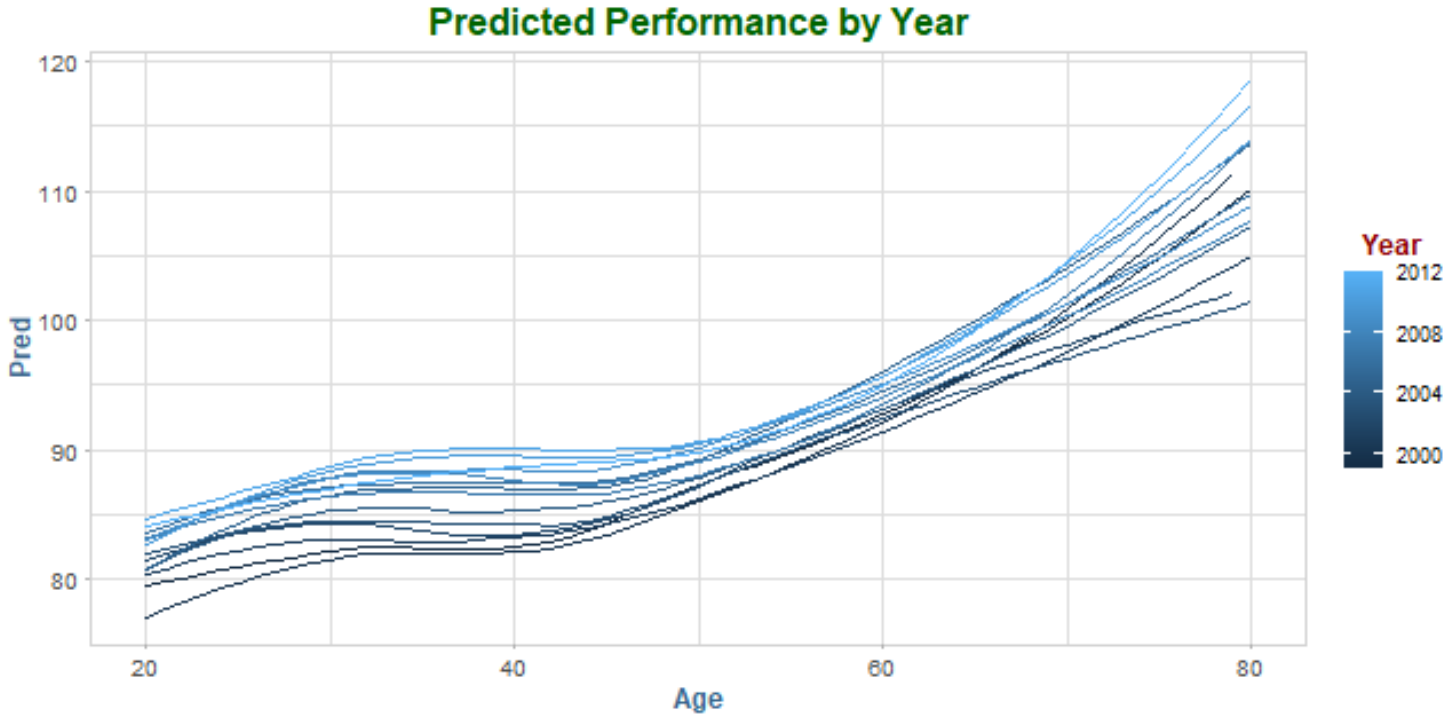
```
plot(mR.lo.pr99 ~ age20to80,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Fitted Run Time (minutes)")

lines(x = age20to80, y = mR.lo.pr12,
      col = "green", lty = 2, lwd = 3)

legend("topleft", col = c("purple", "green"), lty = 1:2, lwd = 3,
       legend =  c("1999", "2012"), bty = "n")
```




```
years <- 1999:2012
results <- list(length(years))

y <- 1999

for( i in 1:length(years) )
{
   y = years[i]
   data <- cbMenSub[ which(cbMenSub$year == y), ]

   model <- loess(runTime ~ age, data)
```

```
   pred <- predict(model, newdata = age20to80)

   results[[i]] <- data.table(Year = rep(y, length(pred)), Age = age20to80, Pred = pred)
}

race_years_data <- do.call("rbind", results)

ggplot(race_years_data, aes(Age, Pred, group = Year)) +
   geom_line(aes(col = Year)) +
   labs(title = "Predicted Performance by Year", ylab = "Predicted Run Time", xlab = "Runner Ag
```

```
Warning: Removed 2 row(s) containing missing values (geom_path).
```



Constructing a Record for an Individual Runner across Years

```
trimBlanks <- function(charVector) {
   nameClean <- gsub("^[[:blank:]]+", "", charVector)
   nameClean <- gsub("[[:blank:]]+$", "", nameClean)
   nameClean <- gsub("[[:blank:]]+", " ", nameClean)
}

nameClean <- trimBlanks(cbMenSub$name)
```

```
length(nameClean)
```

```
[1] 69735
```

```
length(unique(nameClean))
```

```
[1] 42838
```

```
table(table(nameClean))
```

```

    1     2     3     4     5     6     7     8     9    10    11    12    13
29258  7711  2733  1385   712   416   248   149    92    56    44    19     7
   14    15    17    18    19    30
    3     1     1     1     1     1
```

```
head(sort(table(nameClean), decreasing = T), 1)
```

```
nameClean
Michael Smith
           30
```

```
mSmith <- cbMenSub[nameClean == "Michael Smith", ]

head(unique(mSmith$home))
```

```
[1] "Annapolis MD       "  "Bethesda MD        "  " Annapolis MD       "
[4] " Chevy Chase MD     " " Annandale VA       " "Annapolis MD        "
```

```
nameClean <- tolower(nameClean)

head(sort(table(nameClean), decreasing = T), 1)
```

```
nameClean
michael smith
           33
```

```
nameClean <- gsub("[,.]", "", nameClean)
```

```
tabNameYr <- table(cbMenSub$year, nameClean)

max(tabNameYr)
```

```
[1] 5
```

```
class(tabNameYr)
```

```
[1] "table"
```

```
mode(tabNameYr)
```

```
[1] "numeric"
```

```
names(attributes(tabNameYr))
```

```
[1] "dim"      "dimnames" "class"
```

```
dim(tabNameYr)
```

```
[1]    14 39096
```

```
head(colnames(tabNameYr), 3)
```

```
[1] "8illiam maury"   "a gudu memon"    "a miles simmons"
```

```
which(tabNameYr == max(tabNameYr), arr.ind = T)
```

```
        nameClean
2012 14     25444
```

```
indMax <- which(tabNameYr == max(tabNameYr), arr.ind = T)
colnames(tabNameYr)[indMax[2]]
```

```
[1] "michael brown"
```

```
cbMenSub$nameClean <- nameClean
```

```
cbMenSub$yob <- cbMenSub$year - cbMenSub$age

homeClean <- trimBlanks(cbMenSub$home)
homeClean <- tolower(homeClean)

cbMenSub$homeClean <- homeClean
```

```
vars <- c("year", "homeClean", "nameClean", "yob", "runTime")
mb <- which(nameClean == "michael brown")
birthOrder <- order(cbMenSub$yob[mb])
cbMenSub[mb[birthOrder], vars]
```

```
          year      homeClean     nameClean  yob   runTime
2000.2514 2000      tucson az michael brown 1939  96.88333
2010.4230 2010  north east md michael brown 1953  92.26667
2011.3025 2011  north east md michael brown 1953  85.95000
2012.3800 2012  north east md michael brown 1953  88.43333
2009.5237 2009      oakton va michael brown 1957  99.73333
2008.3895 2008     ashburn va michael brown 1958  93.73333
2009.3500 2009     ashburn va michael brown 1958  88.56667
2010.5298 2010     ashburn va michael brown 1958  99.75000
2012.4078 2012      reston va michael brown 1958  89.95000
2006.2625 2006    chevy chase michael brown 1966  84.56667
2010.1896 2010 chevy chase md michael brown 1966  79.35000
```

```
2012.5089 2012 chevy chase md michael brown 1966  95.81667
2004.998  2004  berryville va michael brown 1978  76.31667
2008.2501 2008   arlington va michael brown 1984  84.68333
2010.6296 2010    new york ny michael brown 1984 110.88333
2011.2273 2011   arlington va michael brown 1984  81.70000
2012.881  2012   arlington va michael brown 1984  70.93333
2012.3084 2012     clifton va michael brown 1988  84.88333
```

```
cbMenSub$ID = paste(nameClean, cbMenSub$yob, sep = "_")

races <- tapply(cbMenSub$year, cbMenSub$ID, length)

races8 <- names(races)[which(races >= 8)]

men8 <- cbMenSub[ cbMenSub$ID %in% races8, ]

orderByRunner <- order(men8$ID, men8$year)

men8 <- men8[orderByRunner, ]
```

```
men8L <- split(men8, men8$ID)
names(men8L)
```

```
  [1] "aaron glahe_1974"           "abiy zewde_1967"
  [3] "adam bain_1962"             "adam hughes_1978"
  [5] "adam knapp_1977"            "adam stolzberg_1976"
  [7] "al navidi_1960"             "alan kraut_1951"
  [9] "alan pemberton_1953"        "alan rider_1936"
 [11] "alan stiffler_1962"         "alexander packard_1970"
 [13] "alfred del grosso_1954"     "allan arbogast_1956"
 [15] "allen greenberg_1966"       "alvin white_1956"
 [17] "amir alibabaie_1962"        "andrew aitken_1962"
 [19] "andrew bernstein_1973"      "andrew klemas_1964"
 [21] "andrew mclaren_1949"        "andrew polott_1956"
 [23] "anthony flowe_1958"         "arthur scott_1960"
 [25] "arya akmal_1968"            "augustine paik_1955"
 [27] "bailey st clair_1939"       "barry bupp_1948"
 [29] "barry goldmeier_1965"       "barry goldsmith_1950"
 [31] "barry smith_1953"           "barton bland_1970"
 [33] "benjamin richter_1957"      "bennett beach_1950"
 [35] "bernard kelly_1956"         "bill maccormack_1943"
 [37] "bill rodgers_1948"          "bill sollers_1940"
 [39] "bill vesey_1949"            "bob brammer_1953"
 [41] "bob kramer_1950"            "brad seibert_1957"
 [43] "brandon dubois_1966"        "brian byrne_1948"
 [45] "brian carroll_1956"         "brian chabot_1965"
```

```
 [47] "brian kass_1969"           "brian lane_1975"
 [49] "brian murphy_1953"         "brian robertson_1950"
 [51] "bruce kirch_1960"          "bruce whitson_1946"
 [53] "carl ek_1948"              "carl lay_1944"
 [55] "charles banks_1961"        "charles both_1944"
 [57] "charles clark_1936"        "charles crout_1967"
 [59] "charles divan_1951"        "charles sardo_1954"
 [61] "charles taylor_1947"       "charlie sole_1946"
 [63] "chris ebert_1957"          "chris quasebarth_1960"
 [65] "chris riley_1944"          "chris sega_1955"
 [67] "christian arriola_1976"    "christoph duenwald_1966"
 [69] "christopher jones_1973"    "christopher miller_1973"
 [71] "christopher sten_1944"     "chuck naegeli_1949"
 [73] "clinton schmitt_1957"      "colm dunne_1974"
 [75] "craig berkey_1968"         "craig witmer_1961"
 [77] "curt allen_1957"           "curtis dalton_1952"
 [79] "dale anderson_1975"        "dale jordan_1953"
 [81] "dale learn_1970"           "dallas harrison_1966"
 [83] "daniel barton_1979"        "daniel keany_1956"
 [85] "daryl deprenger_1950"      "daryl knuth_1956"
 [87] "david andrews_1957"        "david cascio_1963"
 [89] "david chernicky_1952"      "david downin_1946"
 [91] "david farrisee_1957"       "david fleming_1954"
 [93] "david gearin_1945"         "david lambert_1960"
 [95] "david landau_1956"         "david mead_1969"
 [97] "david pearson_1961"        "david phillips_1970"
 [99] "david poole_1968"          "david sahnow_1963"
[101] "david walker_1944"         "david wiesenhahn_1962"
[103] "dean siedlecki_1957"       "dennis barr_1954"
[105] "dennis faust_1942"         "dennis loy_1950"
[107] "denny gainer_1952"         "desi alston_1953"
[109] "dick stark_1957"           "dick woods_1947"
[111] "dj waldow_1976"            "donald hensel_1945"
[113] "donald warren_1953"        "douglas dunlop_1954"
[115] "douglas edgecomb_1970"     "douglas klein_1959"
[117] "douglas lunenfeld_1971"    "dov lutzker_1971"
[119] "duane ingalsbe_1940"       "earle fingerhut_1943"
[121] "edward bacon_1954"         "edward green_1932"
[123] "edward hagarty_1955"       "edward hollander_1941"
[125] "edward jefferson_1934"     "edward kopeck_1947"
[127] "edward neighbour_1963"     "elliott hamilton_1961"
[129] "emmet davitt_1958"         "eric katkow_1945"
[131] "eric melby_1949"           "eric winslow_1966"
[133] "erik fatemi_1966"          "erin mccartney_1973"
[135] "eugene elrod_1950"         "eugene kenney_1947"
```

```
[137] "evan roberts_1971"          "forest sun_1955"
[139] "francisco cordova_1960"     "frank jankoski_1954"
[141] "frank manganiello_1946"     "frank myers_1962"
[143] "frank surface_1963"         "fred carson_1940"
[145] "gabriel gluck_1948"         "gary anderson_1953"
[147] "gary chidester_1948"        "gary kodeck_1952"
[149] "gary presuhn_1955"          "gene grady_1949"
[151] "george englert_1949"        "george poporad_1950"
[153] "george yannakakis_1932"     "gerald brown_1957"
[155] "gerald royce_1942"          "gerard lacourciere_1968"
[157] "gilbert macias_1955"        "glenn geelhoed_1942"
[159] "graham anderson_1964"       "grant stewart_1971"
[161] "gregg hinkle_1963"          "guillermo cabrera_1971"
[163] "gustavo olmedo_1958"        "hal danoff_1960"
[165] "harold rosen_1943"          "harrison grayson_1952"
[167] "houng soo_1950"             "howard scruggs_1959"
[169] "hunter montgomery_1969"     "ian heavers_1979"
[171] "ira leibowitz_1952"         "jack tosi_1960"
[173] "jaime salcedo_1953"         "james blackwood_1985"
[175] "james carey_1952"           "james christina_1958"
[177] "james ferguson_1967"        "james harden_1964"
[179] "james mcnabb_1966"          "james mort_1947"
[181] "james peischel_1967"        "james scarborough_1959"
[183] "james snee_1961"            "james trump_1956"
[185] "james weiss_1975"           "jamie hoag_1977"
[187] "jan cook_1965"              "jason tripp_1973"
[189] "jay jacob wind_1950"        "jeffrey allen_1962"
[191] "jeffrey gutman_1950"        "jeffrey kempic_1963"
[193] "jerry marty_1947"           "jim ashworth_1963"
[195] "jim bradford_1963"          "jim cavanaugh_1942"
[197] "jim doyle_1952"             "jim katzman_1967"
[199] "jim noone_1945"             "jim o'donnell_1964"
[201] "john baxter_1947"           "john beard_1960"
[203] "john dean_1945"             "john dix_1949"
[205] "john duda_1962"             "john faith_1949"
[207] "john flynn_1956"            "john haubert_1943"
[209] "john ianno_1956"            "john maclean_1953"
[211] "john mariani_1963"          "john mctyre_1954"
[213] "john miller_1954"           "john moeller_1958"
[215] "john pace_1965"             "john pappajohn_1964"
[217] "john sauer_1956"            "john solberg_1953"
[219] "john sonntag_1968"          "john stanmore_1963"
[221] "john thorsen_1936"          "john tobe_1961"
[223] "john wheatland_1949"        "john winkert_1957"
[225] "jon handel_1977"            "jon laurich_1949"
```

```
[227] "jon reinhard_1963"          "jon wolfsthal_1967"
[229] "jonathan agin_1972"         "jonathan mcmullen_1952"
[231] "jorge paredes_1967"         "joseph durso_1955"
[233] "joseph khalil_1966"         "joseph mccloskey_1947"
[235] "joseph mirarchi_1967"       "joseph valenza_1947"
[237] "joseph vida_1971"           "joseph white_1939"
[239] "julian angelone_1944"       "keith buell_1974"
[241] "ken buja_1964"              "ken krehbiel_1954"
[243] "ken landauer_1960"          "ken quincy_1938"
[245] "kenneth cockerill_1962"     "kenneth kelley_1939"
[247] "kenyon erickson_1954"       "kevin adams_1957"
[249] "kevin barrett_1959"         "kevin cassidy_1963"
[251] "kevin keany_1955"           "kevin kunkel_1972"
[253] "kevin mcmahon_1960"         "kevin moore_1973"
[255] "kevin tullier_1971"         "kevin walsh_1963"
[257] "kevin yates_1972"           "kurt landauer_1954"
[259] "kurt quasebarth_1963"       "kyle barton_1979"
[261] "larry denino_1961"          "lary larson_1955"
[263] "laszlo madaras_1962"        "lee youngblood_1955"
[265] "len doughty_1958"           "len gemma_1960"
[267] "leonard bechtel_1964"       "leonard lee_1966"
[269] "les graber_1962"            "les pang_1953"
[271] "lester brown_1934"          "lewis parker_1965"
[273] "loren bussert_1947"         "lou lodovico_1924"
[275] "louis demouy_1940"          "louis garczynski_1940"
[277] "luis amaya_1963"            "lyle jentzer_1953"
[279] "mac mcneil_1949"            "madis muller_1977"
[281] "malcolm poulin_1957"        "marc de angelis_1957"
[283] "marc gunther_1951"          "marc wolfson_1950"
[285] "mark davies_1970"           "mark fraley_1957"
[287] "mark freeman_1975"          "mark hoon_1965"
[289] "mark johnson_1963"          "mark kline_1952"
[291] "mark lin_1966"              "mark lippman_1962"
[293] "mark malander_1958"         "mark neff_1962"
[295] "mark palmer_1967"           "mark wisch_1959"
[297] "mark wolff_1961"            "martin mclean_1945"
[299] "marvin pace_1954"           "matthew chesnes_1979"
[301] "matthew gaertner_1974"      "matthew haskins_1968"
[303] "michael coffee_1972"        "michael davitt_1954"
[305] "michael dusenbery_1979"     "michael gadbaw_1948"
[307] "michael glikes_1968"        "michael glover_1961"
[309] "michael golash_1944"        "michael greenwalt_1959"
[311] "michael greer_1959"         "michael hedrick_1972"
[313] "michael kellogg_1955"       "michael lustig_1967"
[315] "michael martin_1951"        "michael mashner_1980"
```

```
[317] "michael matyas_1955"        "michael mclenigan_1964"
[319] "michael mcroberts_1963"     "michael rayne_1956"
[321] "michael rosenthal_1968"     "michael scheurer_1949"
[323] "michael scott_1957"         "michael smith_1963"
[325] "michael stievater_1959"     "michael triantafillou_1960"
[327] "mike acuna_1966"            "mike duncan_1956"
[329] "milton vazquez_1954"        "mitchell krasnopoler_1958"
[331] "mohammed zaatari_1967"      "nathan greenbaum_1949"
[333] "neil levin_1957"            "neil shepherd_1958"
[335] "neil simons_1959"           "nianxiang xie_1928"
[337] "nicholas clark_1951"        "norm coleman_1945"
[339] "omar ali_1971"              "pat piscitelli_1956"
[341] "patrick connors_1982"       "patrick griffith_1945"
[343] "patrick hinderdael_1959"    "patrick kunze_1980"
[345] "paul aloe_1957"             "paul bousel_1954"
[347] "paul brown_1967"            "paul durbin_1965"
[349] "paul elias_1962"            "paul fiondella_1947"
[351] "paul foster_1956"           "paul garrard_1956"
[353] "paul grosz_1950"            "paul kates_1970"
[355] "paul loebach_1969"          "paul sandy_1960"
[357] "paul schlereth_1951"        "paul sharratt_1956"
[359] "paul warren_1950"           "paul wilder_1966"
[361] "peter comfort_1950"         "peter farley_1973"
[363] "peter hemphill_1959"        "peter horton_1956"
[365] "peter lunt_1950"            "peter mckeen_1959"
[367] "peter reilly_1957"          "philip rizzi_1964"
[369] "pierce mcmanus_1971"        "pierre donahue_1963"
[371] "prasad gerard_1959"         "radhakisan baheti_1945"
[373] "rahul sood_1985"            "ralph mckinney_1945"
[375] "ray celeste_1959"           "ray lake_1960"
[377] "reginald trujillo_1949"     "rich luquette_1952"
[379] "richard barton_1948"        "richard behnke_1945"
[381] "richard carter_1972"        "richard fox_1948"
[383] "richard glenn_1969"         "richard joseph_1964"
[385] "richard kaplar_1952"        "richard mitchell_1945"
[387] "richard williams_1936"      "rick berzon_1953"
[389] "rick kern_1966"             "rick pulley_1958"
[391] "rick westley_1948"          "robert cassagnol_1956"
[393] "robert daniels_1961"        "robert falk_1964"
[395] "robert gray_1956"           "robert hall_1972"
[397] "robert keith_1949"          "robert platt_1952"
[399] "robert roche_1954"          "robert smith_1934"
[401] "robert trost_1947"          "robert vaughn_1940"
[403] "robert walker_1968"         "roger kuehnle_1956"
[405] "roger minor_1953"           "ron wolak_1947"
```

```
[407] "ronald busch_1961"          "ronnie wong_1947"
[409] "roy beaumont_1972"          "roy cargiulo_1962"
[411] "russ cooke_1952"            "samuel floyd_1943"
[413] "samuel richman_1958"        "samuel wyman_1972"
[415] "scott bell_1959"            "scott hall_1967"
[417] "scott hubert_1958"          "scott hunt_1968"
[419] "scott kohr_1962"            "scott koonce_1972"
[421] "sean hicks_1973"            "sean keely_1972"
[423] "sean logan_1963"            "sean rhoderick_1973"
[425] "sid kaplan_1947"            "stephen chavez_1953"
[427] "stephen forman_1941"        "stephen johnson_1945"
[429] "stephen koch_1961"          "stephen mostow_1971"
[431] "stephen silberstein_1959"   "stephen svab_1955"
[433] "steve fulton_1952"          "steven grufferman_1953"
[435] "steven kaplow_1955"         "steven maguire_1967"
[437] "steven palkovitz_1961"      "steven teslik_1954"
[439] "ted poulos_1962"            "terry schnarrs_1958"
[441] "thip vongxay_1979"          "thomas askins_1955"
[443] "thomas engle_1959"          "thomas momiyama_1932"
[445] "thomas simpson_1943"        "thomas skelly_1952"
[447] "tim appenzeller_1960"       "tim kirkner_1962"
[449] "tim rowe_1955"              "timothy mcquade_1963"
[451] "timothy morgan_1951"        "timothy oldham_1947"
[453] "todd kane_1951"             "tom ivey_1959"
[455] "tom jones_1958"             "tom ray_1934"
[457] "tom stone_1966"             "tom tobin_1955"
[459] "tom winkert_1965"           "tony santucci_1954"
[461] "tony zukas_1954"            "tracy wilson_1960"
[463] "tyler jug_1971"             "vasilios stayeas_1947"
[465] "victor finnegan_1957"       "w ralph eubanks_1958"
[467] "walter winans_1952"         "warren baise_1950"
[469] "william brooks_1952"        "william cavanaugh_1942"
[471] "william cho_1958"           "william clem_1960"
[473] "william furlong_1955"       "william haskins_1970"
[475] "william noonan_1972"        "william parsons_1948"
[477] "william pulver_1938"        "william scott_1948"
[479] "william waskes_1977"        "yonghee nam_1961"
```

```
length(unique(men8$ID))
```

```
[1] 480
```

```
length(men8L)
```

```
[1] 480
```

```
gapTime <- tapply(men8$runTime, men8$ID,
                  function(t) any(abs(diff(t)) > 20))

gapTime <- sapply(men8L, function(df)
   any(abs(diff(df$runTime)) > 20))

sum(gapTime)
```

```
[1] 49
```

```
lapply(men8L[ gapTime ][1:2], function(df) df[, vars])
```

```
$`abiy zewde_1967`
          year            homeClean  nameClean  yob   runTime
1999.2640 1999      gaithersburg md abiy zewde 1967  96.51667
2000.2604 2000   montgomery vill md abiy zewde 1967  96.63333
2001.2261 2001   montgomery vill md abiy zewde 1967  89.10000
2002.3684 2002   montgomery vill md abiy zewde 1967 123.00000
2003.3293 2003      gaithersburg md abiy zewde 1967  97.68333
2004.3579 2004   montgomery vill md abiy zewde 1967 100.36667
2006.4833 2006         gaithersburg abiy zewde 1967 108.40000
2008.4560 2008   montgomery vill md abiy zewde 1967  98.78333
2009.5063 2009 montgomery villag md abiy zewde 1967  98.50000
2010.5319 2010 montgomery villag md abiy zewde 1967  99.91667
2011.6492 2011 montgomery villag md abiy zewde 1967 113.10000
2012.3085 2012 montgomery villag md abiy zewde 1967  84.88333

$`adam hughes_1978`
          year     homeClean   nameClean  yob   runTime
2005.1916 2005 washington dc adam hughes 1978  80.38333
2006.2273 2006    washington adam hughes 1978  85.16667
2007.1362 2007 washington dc adam hughes 1978  77.78333
2008.1028 2008 washington dc adam hughes 1978  74.23333
2009.5971 2009 washington dc adam hughes 1978 108.06667
2010.5630 2010 washington dc adam hughes 1978 103.06667
2011.1484 2011 washington dc adam hughes 1978  77.11667
2012.1836 2012 washington dc adam hughes 1978  77.76667
```

```
homeLen <- nchar(cbMenSub$homeClean)

cbMenSub$state <- substr(cbMenSub$homeClean,
                         start = homeLen - 1, stop = homeLen)

cbMenSub$state[cbMenSub$year == 2006] = NA
```

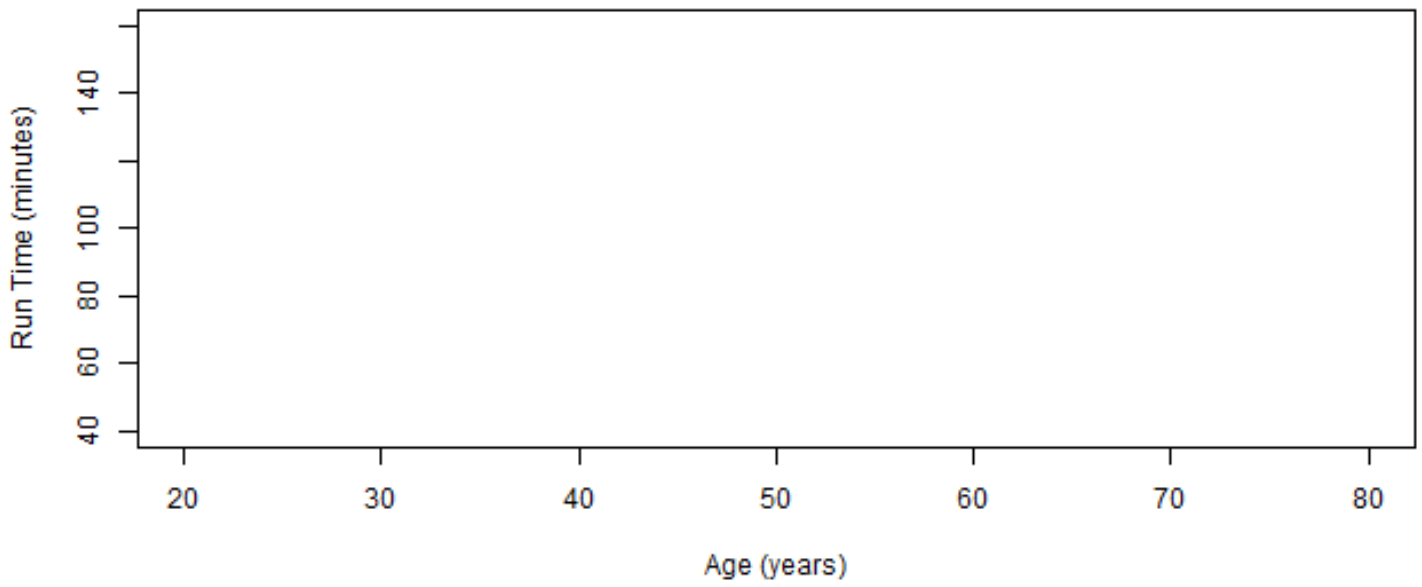
```
cbMenSub$ID = paste(cbMenSub$nameClean, cbMenSub$yob,
                    cbMenSub$state, sep = "_")
```

```
numRaces <- tapply(cbMenSub$year, cbMenSub$ID, length)
races8 <- names(numRaces)[which(numRaces >= 8)]
men8 <- cbMenSub[ cbMenSub$ID %in% races8, ]

orderByRunner <- order(men8$ID, men8$year)
men8 <- men8[orderByRunner, ]

men8L <- split(men8, men8$ID)
names(men8L) <- races8

length(men8L)
```

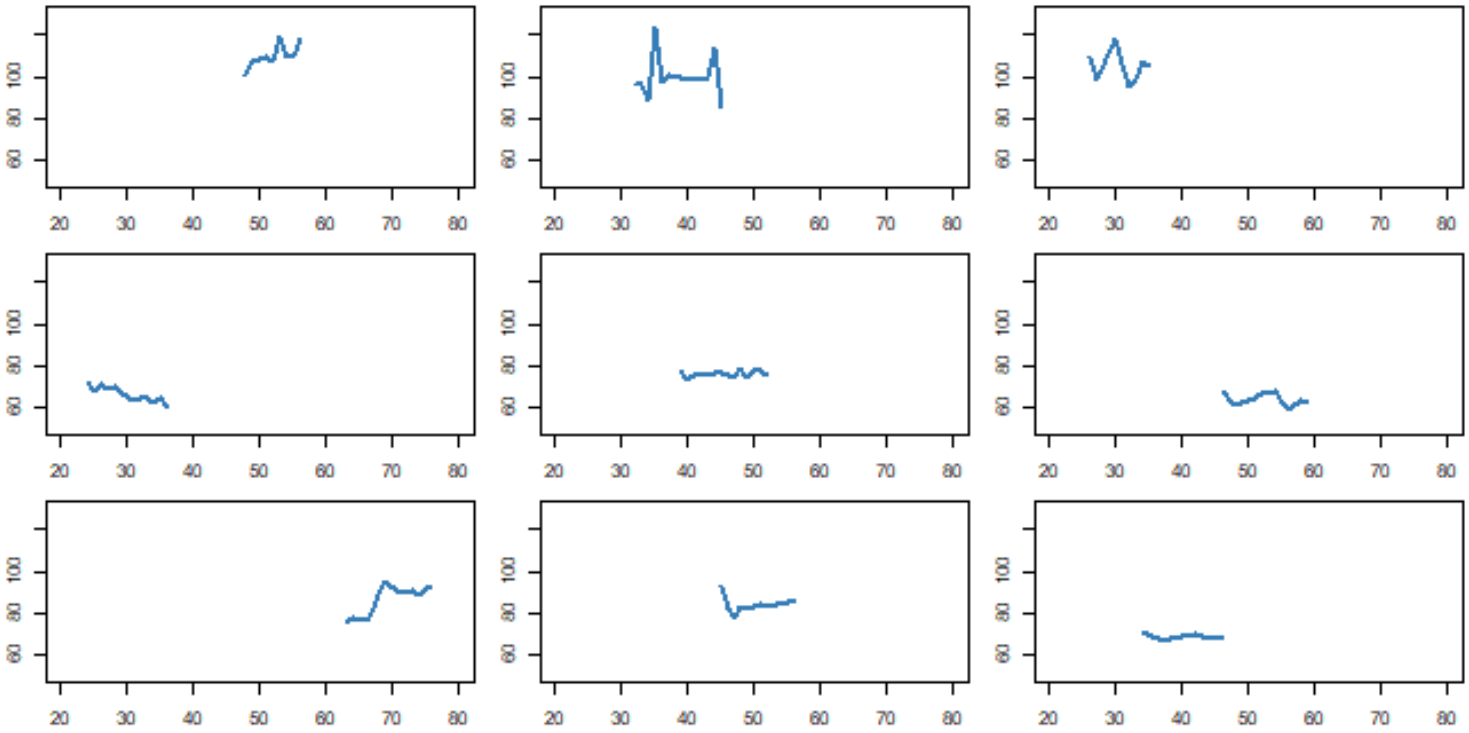
```
[1] 306
```

Modeling the Change in Running Time for Individuals

```
groups <- 1 + (1:length(men8L)) %% 9)

plot(x = 40, y = 60, type = "n",
     xlim = c(20, 80), ylim = c(40, 160),
     xlab = "Age (years)", ylab = "Run Time (minutes)")
```



```
addRunners <- function(listRunners, colors, numLty) {
   numRunners <- length(listRunners)

   colIndx <- 1 + (1:numRunners) %% length(colors)
   ltys <- rep(1:numLty, each = length(colors), length = numRunners)

   mapply(function(df, i) {
      lines(df$runTime ~ df$age,
            col = colors[colIndex[i]], lwd = 2, lty = ltys[i])
   }, listRunners, i = 1:numRunners)
}

colors <- c("#e41alc", "#377eb8", "#4daf4a", "#984ea3",
            "#ff7f00", "#a65628")

par(mfrow = c(3, 3), mar = c(2, 2, 1, 1))
invisible(
   sapply(1:9, function(grpId) {
      plot(x = 0, y = 0, type = "n",
           xlim = c(20, 80), ylim = c(50, 130),
           xlab = "Age (years)", ylab = "Run Time (minutes)")

      addRunners(men8L[ groups == grpId ], colors, numLty = 6)
   })
)
```



```
fitOne <- function(oneRunner, addLine = F, col = "grey") {
   lmOne <- lm(runTime ~ age, data = oneRunner)

   if ( addLine )
      lines(x = oneRunner$age, y = predict(lmOne),
            col = col, lwd = 2, lty = 2)

   ind <- floor( (nrow(oneRunner) + 1) / 2)
   res <- c(coefficients(lmOne)[2], oneRunner$age[ind],
            predict(lmOne)[ ind ])

   names(res) <- c("ageCoeff", "medAge", "predRunTime")

   return(res)
}
```

```
par(mfrow = c(1, 1))
plot( x = 0, y = 0, type = "n",
      xlim = c(20, 80), ylim = c(50, 130),
      xlab = "Age (years)", ylab = "Run Time (minutes)")

addRunners( men8L[ groups == 9 ], colors, numLty = 6)
```

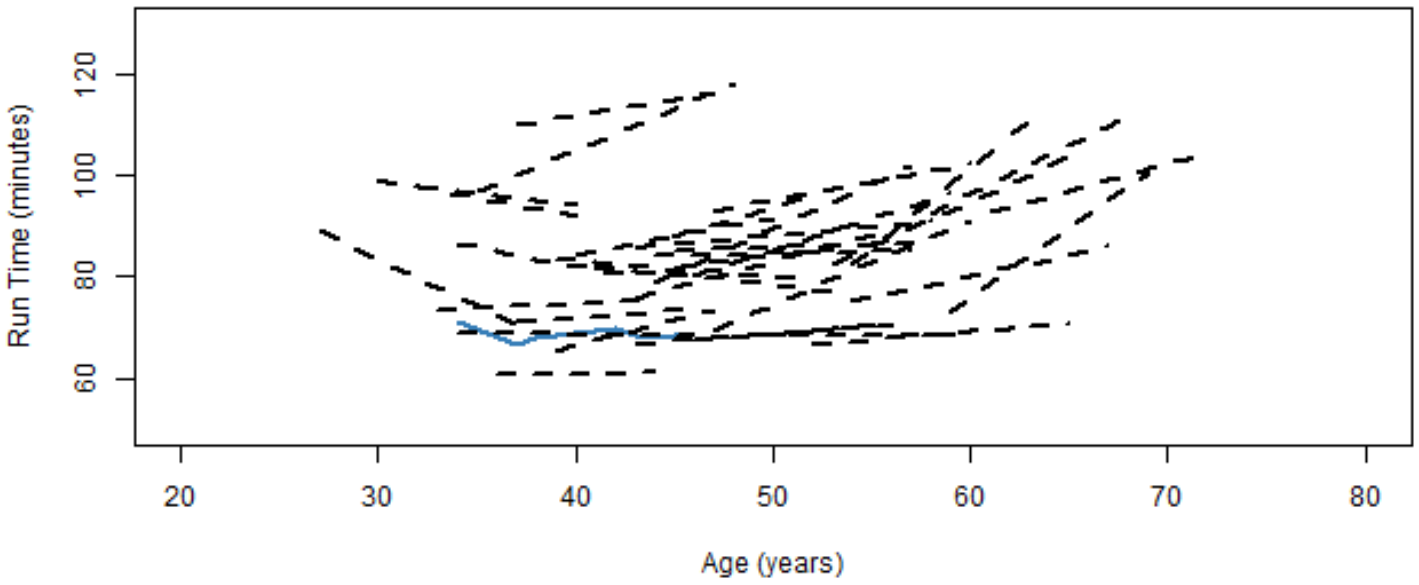
```
$`allen greenberg_1966_dc`
NULL
```

```
$`barry goldmeier_1965_md`
NULL

$`brian carroll_1956_md`
NULL

$`charlie sole_1946_va`
NULL

$`curtis dalton_1952_md`
NULL

$`david gearin_1945_va`
NULL

$`desi alston_1953_va`
NULL

$`edward hagarty_1955_md`
NULL

$`erik fatemi_1966_va`
NULL

$`fred carson_1940_md`
NULL

$`gerald royce_1942_va`
NULL

$`hunter montgomery_1969_md`
NULL

$`james snee_1961_md`
NULL

$`jim o'donnell_1964_dc`
NULL

$`john sauer_1956_md`
NULL

$`jonathan agin_1972_va`
NULL
```

```
$`keith buell_1974_va`
NULL

$`kevin barrett_1959_ma`
NULL

$`len gemma_1960_md`
NULL

$`louis garczynski_1940_va`
NULL

$`mark fraley_1957_oh`
NULL

$`michael davitt_1954_md`
NULL

$`michael mcroberts_1963_va`
NULL

$`milton vazquez_1954_md`
NULL

$`omar ali_1971_md`
NULL

$`paul warren_1950_ny`
NULL

$`ralph mckinney_1945_de`
NULL

$`richard joseph_1964_ny`
NULL

$`robert platt_1952_va`
NULL

$`ronnie wong_1947_md`
NULL

$`stephen chavez_1953_md`
NULL
```

```
$`thomas engle_1959_va`
NULL

$`tracy wilson_1960_va`
NULL

$`william furlong_1955_va`
NULL
```

```
lapply(men8L[groups == 9], fitOne, addLine = T, col = "black")
```




```
$`allen greenberg_1966_dc`
   ageCoeff      medAge predRunTime
-0.06422567 42.00000000 68.62088587

$`barry goldmeier_1965_md`
   ageCoeff      medAge predRunTime
  0.6043803  43.0000000 113.6788462

$`brian carroll_1956_md`
   ageCoeff      medAge predRunTime
  0.2984209  51.0000000  69.1152850

$`charlie sole_1946_va`
   ageCoeff      medAge predRunTime
```

```
    1.74380    56.00000    87.81894

$`curtis dalton_1952_md`
   ageCoeff      medAge predRunTime
  0.7106573  53.0000000  97.3086737

$`david gearin_1945_va`
   ageCoeff      medAge predRunTime
  0.8283955  62.0000000  81.8074764

$`desi alston_1953_va`
   ageCoeff      medAge predRunTime
  0.3102904  50.0000000  68.6017677

$`edward hagarty_1955_md`
   ageCoeff      medAge predRunTime
 0.07279959 50.00000000 84.63434891

$`erik fatemi_1966_va`
   ageCoeff      medAge predRunTime
  0.1793474  37.0000000  74.1732342

$`fred carson_1940_md`
   ageCoeff      medAge predRunTime
   2.748825   63.000000   84.210043

$`gerald royce_1942_va`
   ageCoeff      medAge predRunTime
     1.9102     62.0000    100.1906

$`hunter montgomery_1969_md`
   ageCoeff      medAge predRunTime
 -0.6917469  39.0000000  82.8802184

$`james snee_1961_md`
   ageCoeff      medAge predRunTime
 -0.3151655  46.0000000  81.3345616

$`jim o'donnell_1964_dc`
   ageCoeff      medAge predRunTime
   1.628891   38.000000  101.717910

$`john sauer_1956_md`
   ageCoeff      medAge predRunTime
   1.037477   48.000000   81.073766
```

```
$`jonathan agin_1972_va`
   ageCoeff      medAge predRunTime
 -0.7079431  35.0000000  95.4824610

$`keith buell_1974_va`
   ageCoeff      medAge predRunTime
  -1.836643   33.000000   78.038153

$`kevin barrett_1959_ma`
   ageCoeff      medAge predRunTime
  0.4906609  49.0000000  83.7143678

$`len gemma_1960_md`
   ageCoeff      medAge predRunTime
   0.737585   43.000000   86.526956

$`louis garczynski_1940_va`
   ageCoeff      medAge predRunTime
   1.051165   67.000000   99.192981

$`mark fraley_1957_oh`
   ageCoeff      medAge predRunTime
   1.201748   47.000000   90.286407

$`michael davitt_1954_md`
   ageCoeff      medAge predRunTime
  0.3352041  51.0000000  88.4843537

$`michael mcroberts_1963_va`
   ageCoeff      medAge predRunTime
  0.2701389  40.0000000  72.0236111

$`milton vazquez_1954_md`
   ageCoeff      medAge predRunTime
 0.09781651 53.00000000 86.13745994

$`omar ali_1971_md`
   ageCoeff      medAge predRunTime
 -0.4851026  36.0000000  96.0043878

$`paul warren_1950_ny`
   ageCoeff      medAge predRunTime
-0.01358711 53.00000000 68.39056173
```

```
$`ralph mckinney_1945_de`
   ageCoeff      medAge predRunTime
   3.162451   58.000000   94.960196

$`richard joseph_1964_ny`
   ageCoeff      medAge predRunTime
 0.01013514 39.00000000 60.99864865

$`robert platt_1952_va`
   ageCoeff      medAge predRunTime
   1.668127   52.000000   77.869373

$`ronnie wong_1947_md`
   ageCoeff      medAge predRunTime
  0.3051343  58.0000000  68.5476303

$`stephen chavez_1953_md`
   ageCoeff      medAge predRunTime
   1.228651   51.000000   86.551087

$`thomas engle_1959_va`
   ageCoeff      medAge predRunTime
 -0.4032051  46.0000000  80.0532051

$`tracy wilson_1960_va`
   ageCoeff      medAge predRunTime
    1.02548    42.00000    68.51324

$`william furlong_1955_va`
   ageCoeff      medAge predRunTime
   1.736293   49.000000   87.749356
```

```
men8LongFit <- lapply(men8L, fitOne)

coeffs <- sapply(men8LongFit, "[", "ageCoeff" )
ages <- sapply(men8LongFit, "[", "medAge")

longCoeffs <- lm(coeffs ~ ages)

summary(longCoeffs)
```

```

Call:
lm(formula = coeffs ~ ages)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-4.4026 -0.6375 -0.0246  0.5645  3.3541

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.958440   0.305487  -6.411 5.51e-10 ***
ages         0.055263   0.006175   8.949  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.01 on 304 degrees of freedom
Multiple R-squared:  0.2085,    Adjusted R-squared:  0.2059
F-statistic: 80.09 on 1 and 304 DF,  p-value: < 2.2e-16
```

Further Analysis

1.)

Write a function that uses *read.fwf()* to read the 28 text tables in MenTxt/ and WomenTxt/ into R. These are called 1999.text, 2000txt, etc. and are described in greater detail in 2.2. Examine the tables in a plain text editor to determine the start and end position of each column of interest (name, hometown, age and gun and net time).

Use statistics to explore the results and confirm that you have extracted the information from the correct positions in the text.

```
data_paths <- sapply(c("men_txt", "women_txt"), function(p) {
   path <- file.path(data.dir, p)

   sapply(path, function( f ) {
      file.path(path, list.files(path) )
   })
})

consolidated_files <- c(data_paths[, 1], data_paths[, 2])


tp <- consolidated_files[1]

files <- lapply(consolidated_files, function(f) read.fwf(f, widths = 120))

length(files)
```

```
[1] 28
```

2.)

Revise the extractVariables function (see section 2.2) to remove the rows in *menTables* that are blank. In addition, eliminatee the rows that begin with a '*' or a '#'. You may find the following regular expression helpful for locating blank rows in a table.

grep("[1]*$", body)

The pattern uses several meta characters. The ^ is an anchor for the start of the string, the $ anchors to the end of the string, the [[:blank:]] denotes the equivalence class of any space or tab character, and * indicates that the blank character can appear 0 or more times. All together the pattern [2]*$ matches a string that contains any number of blanks from start to end.

```
extractVariables =
   function(file, varNames = c("name", "home", "ag", "gun", "net", "time"))
   {
      # Find the index of the row with =s
      eqIndex <- grep("^===", file)
      spacerRow <- file[eqIndex]
      headerRow <- tolower(file[ eqIndex - 1 ])
      body <- file[ -(1 : eqIndex) ]

      blank <- grep("^[[:blank:]]*$", body)
      footnote <- grep("^[^\\s]*[\\*]*[#]", body)
      ignore <- union(blank, footnote)

      if(length(ignore))
         body <- body[-ignore]

      # Obtain the starting and ending positions of variables
      searchLocs <- findColLocs(spacerRow)
      locCols <- selectCols(varNames, headerRow, searchLocs)

      Values <- mapply(substr, list(body), start = locCols[1, ],
                       stop = locCols[2, ])
      colnames(Values) <- varNames

      invisible(Values)
   }
```

3.)

Find the record where the time is only 1.5. What happened? Determine how to handle the problem and which function needs to be modified: *extractResTable()*, *extractVariables()*, or *cleanUp()*. In your modifica-

[1] [:blank:]
[2] [:blank:]

tion, include code to provide a warning message about the rows that are being dropped for having a time that is too small.

4.)

Examine the head and tail of the 2006 men's file. Look at both the character matrix in the list called *menResMat* and the character vector in the list called *menFiles* (see Sec 2.2). (Recall that the desired character matrix in *menResMat* and the character vector in *menFiels* both correspond to the element named "2006"). What is wrong with the hometown? Examine the header closely to figure out how this error came about. Modify the *extractVariables()* function to fix the problem.

```
extractVariables =
   function(file, varNames = c("name", "home", "ag", "gun", "net", "time"))
   {
      # Find the index of the row with =s
      eqIndex <- grep("^===", file)
      spacerRow <- file[eqIndex]
      headerRow <- tolower(file[ eqIndex - 1 ])
      body <- file[ -(1 : eqIndex) ]

      blank <- grep("^[[:blank:]]*$", body)
      footnote <- grep("^[^\\s]*[\\*]*[#]", body)
      ignore <- union(blank, footnote)

      if(length(ignore))
         body <- body[-ignore]

      # Obtain the starting and ending positions of variables
      searchLocs <- findColLocs(spacerRow)
      locCols <- selectCols(varNames, headerRow, searchLocs)

      Values <- mapply(substr, list(body), start = locCols[1, ],
                       stop = locCols[2, ])
      colnames(Values) <- varNames

      invisible(Values)
   }
```

5.)

Write the *convertTime()* function described in Section 2.3. This function takes a string where time is in either the format hh:mm:ss or mm:ss. The return value is the time as numeric value of the number of minutes. Design this function to take a character vector with multiple strings and return a numeric vector.

```
convertTime <- function( charTime ) {

   timePieces <- strsplit(charTime, ":")

   timePieces <- sapply(timePieces, as.numeric)

   runTime <- sapply(timePieces,
                  function(x) {
                     if(length(x) == 2) x[1] + x[2]/60
                     else 60 * x[1] + x[2] + x[3]/60
                  })

}
```

6.)

Modify the *createDF()* function in Section 2.3 to handle the formatting problem with the 2006 male file. You will need to carefully inspect the raw text file in order to determien the problem.

```
createDF =
   function(Res, year, sex) {
      useTime <- if(!is.na(Res[1, 'net']) )
                     Res[, 'net']
                  else if( !is.na(Res[1, 'gun']) )
                     Res[ , 'gun']
                  else
                     Res[, 'time']

      useTime <- gsub("[#\\*[:blank:]]", "", useTime)

      Res <- Res[ useTime != "", ]

      runTime <- convertTime(useTime[ useTime != "" ])

      N <- nrow(Res)

      Results <- data.frame( year = rep(year, N),
                             sex = rep(sex, N),
                             name = Res[, 'name'],
                             home = Res[, 'home'],
                             age = as.numeric(Res[, 'ag']),
                             runTime = runTime,
                             stringsAsFactors = F)
```

```
    invisible(Results)
  }
```

7.)

Follow the approach developed in Section 2.2 to read the files for the female runners and then process them using the functions in Section 2.3 to create a data frame for analysis. You may need to generalize the *createDF()* and *extractVariables()* functions to handle additional oddities in the raw text files.

```
womenDF <- mapply(createDF, womenResMat, year = years,
                  sex = rep("F", 14), SIMPLIFY = F)

sapply(womenDF, function(x) sum(is.na(x$runTime)))
```

```
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

8.)

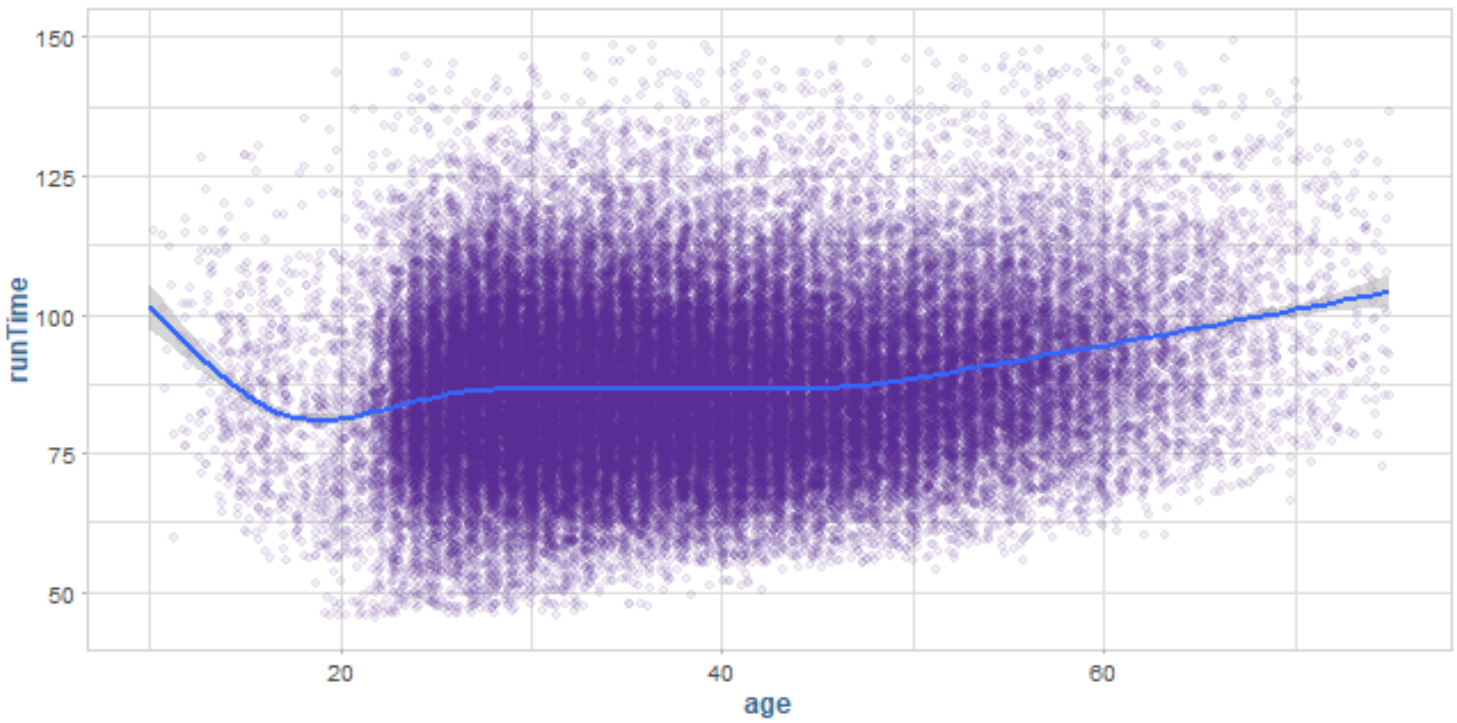
Modify the call to the *plot()* function that created figure 2.6 to create Figure 2.7. To do this, read the documentation for plot() to determine which parameters could be helpful.

```
ggplot(cbMen, aes(age, runTime)) +
  geom_jitter(col = Purple8A) +
  geom_smooth() +
  xlim(10, 75) +
  ylim(45, 150)
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
Warning: Removed 130 rows containing non-finite values (stat_smooth).
```

```
Warning: Removed 137 rows containing missing values (geom_point).
```



9.)

Modify the piecewise linear fit from Section 2.4.2 to include a hing a 70. Examine the coefficients from the fit and compare the fitted curve to the loess curve. Does the additional hing improve the fit?

```
age20to70 <- seq( from = 20, to = 70, by = 1)

mR.lo992 <- loess(runTime ~ age, cbMenSub[ cbMenSub$year == 1999,])
mR.lo.pr992 <- predict(mR.lo99, data.frame(age = age20to70))

summary(mR.lo.pr992)
```

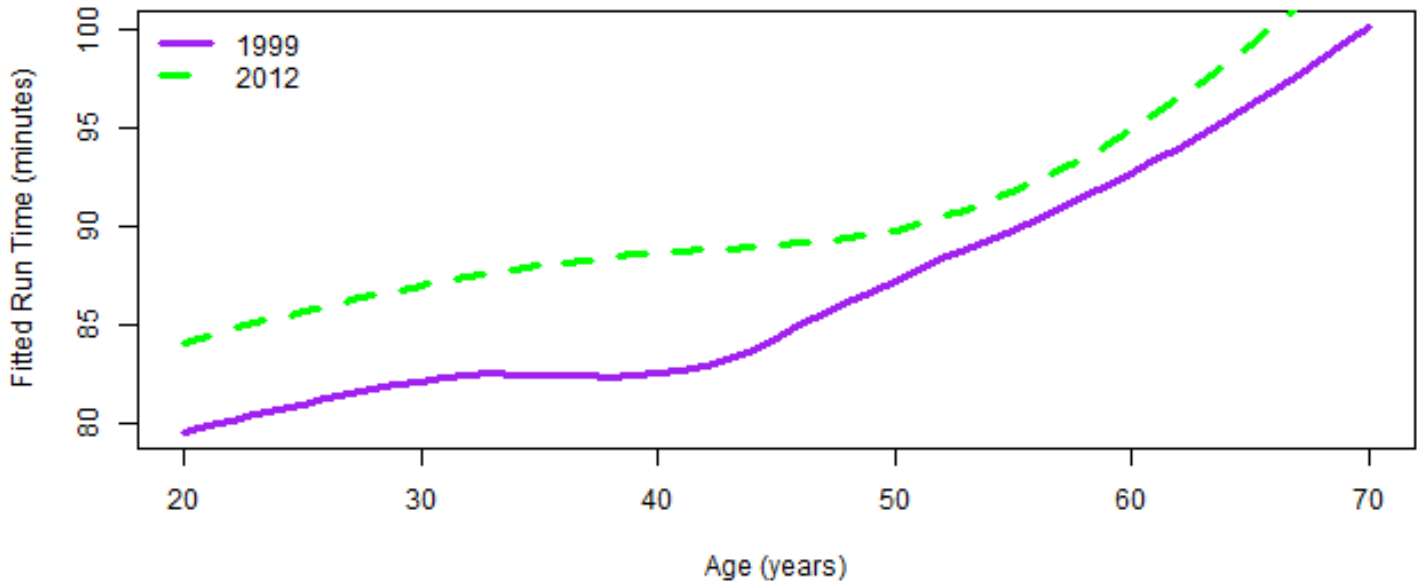
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  79.53   82.38   84.32   86.85   91.18  100.19
```

```
mR.lo122 <- loess(runTime ~ age, cbMenSub[ cbMenSub$year == 2012, ])
mR.lo.pr122 <- predict(mR.lo122, data.frame(age = age20to70))

summary(mR.lo122)
```

```
Call:
loess(formula = runTime ~ age, data = cbMenSub[cbMenSub$year ==
    2012, ])

Number of Observations: 7164
Equivalent Number of Parameters: 5.08
```

```
Residual Standard Error: 15.23
Trace of smoother matrix: 5.55  (exact)

Control settings:
  span      :  0.75
  degree    :  2
  family    :  gaussian
  surface   :  interpolate     cell = 0.2
  normalize:  TRUE
 parametric:  FALSE
drop.square:  FALSE
```

```
plot(mR.lo.pr992 ~ age20to70,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Fitted Run Time (minutes)")

lines(x = age20to70, y = mR.lo.pr122,
      col = "green", lty = 2, lwd = 3)

legend("topleft", col = c("purple", "green"), lty = 1:2, lwd = 3,
       legend =  c("1999", "2012"), bty = "n")
```

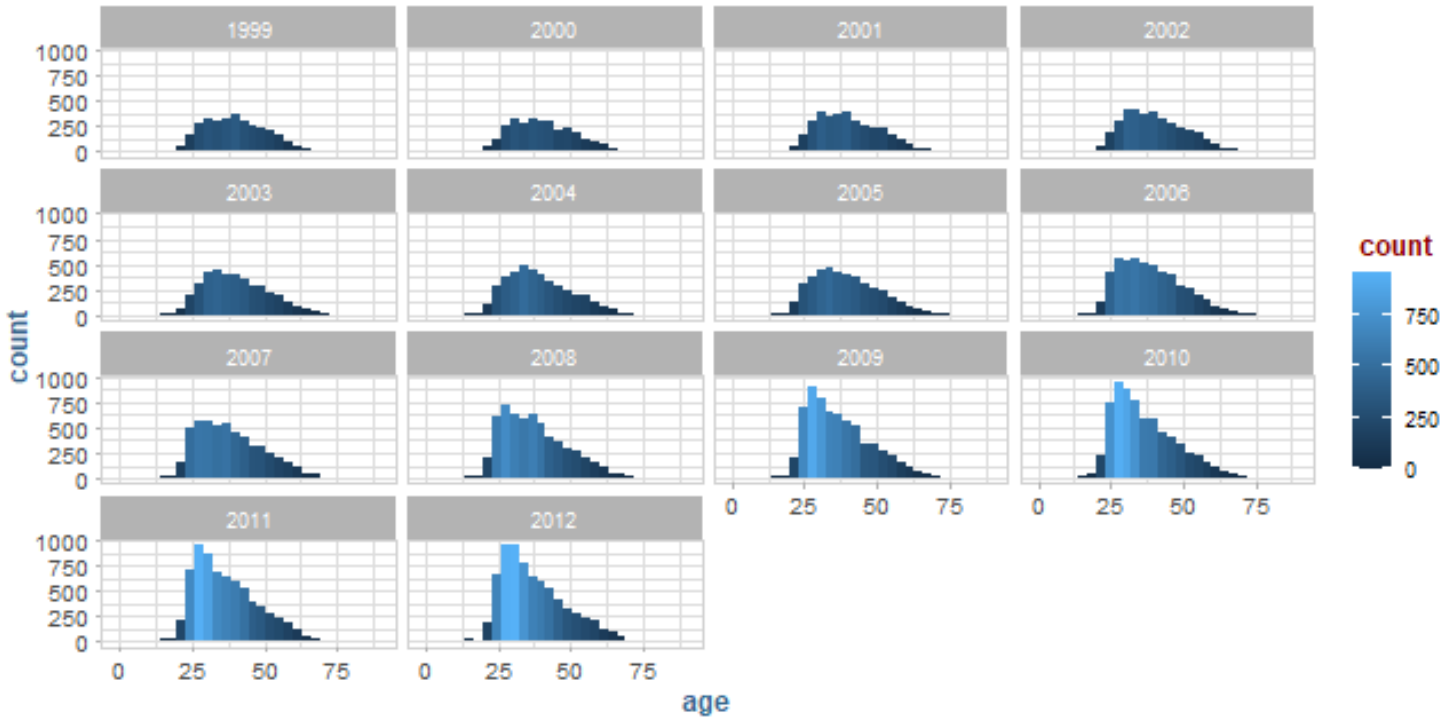


10.)

We have seen that the 1999 runners were typically older than the 2012 runners. Compare the age distribution of the runners agross all 14 years of the races. Use quantile-quantile plots, boxplots and density curves to make your comparisions.

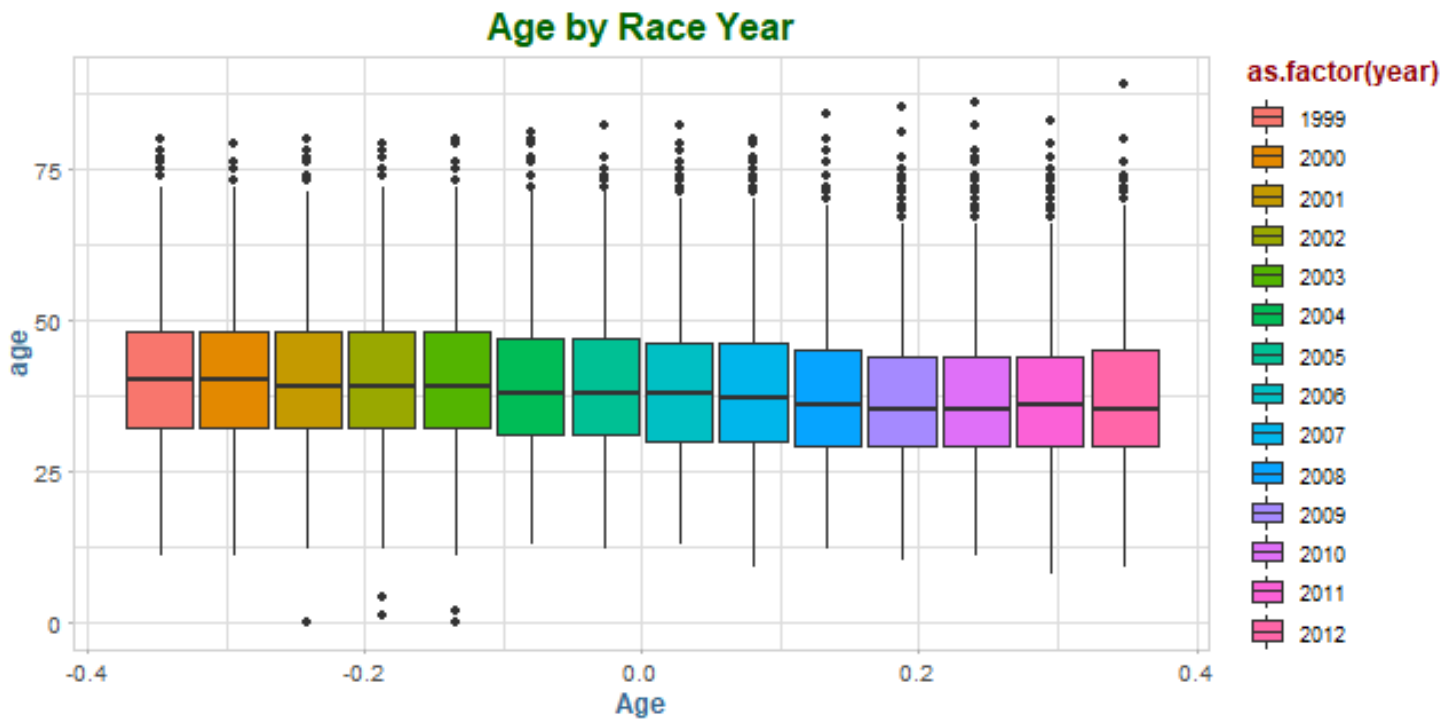
```
ggplot(cbMen, aes(age, group = year)) +
   geom_histogram(aes(fill = ..count..), bins = 30) +
   facet_wrap(~year)
```

```
Warning: Removed 23 rows containing non-finite values (stat_bin).
```




```
ggplot(cbMen, aes(age, group = year)) +
   geom_boxplot(aes(fill = as.factor(year))) +
   coord_flip() +
   labs(title = "Age by Race Year", xlab = "Race Year", ylab = "Age")
```

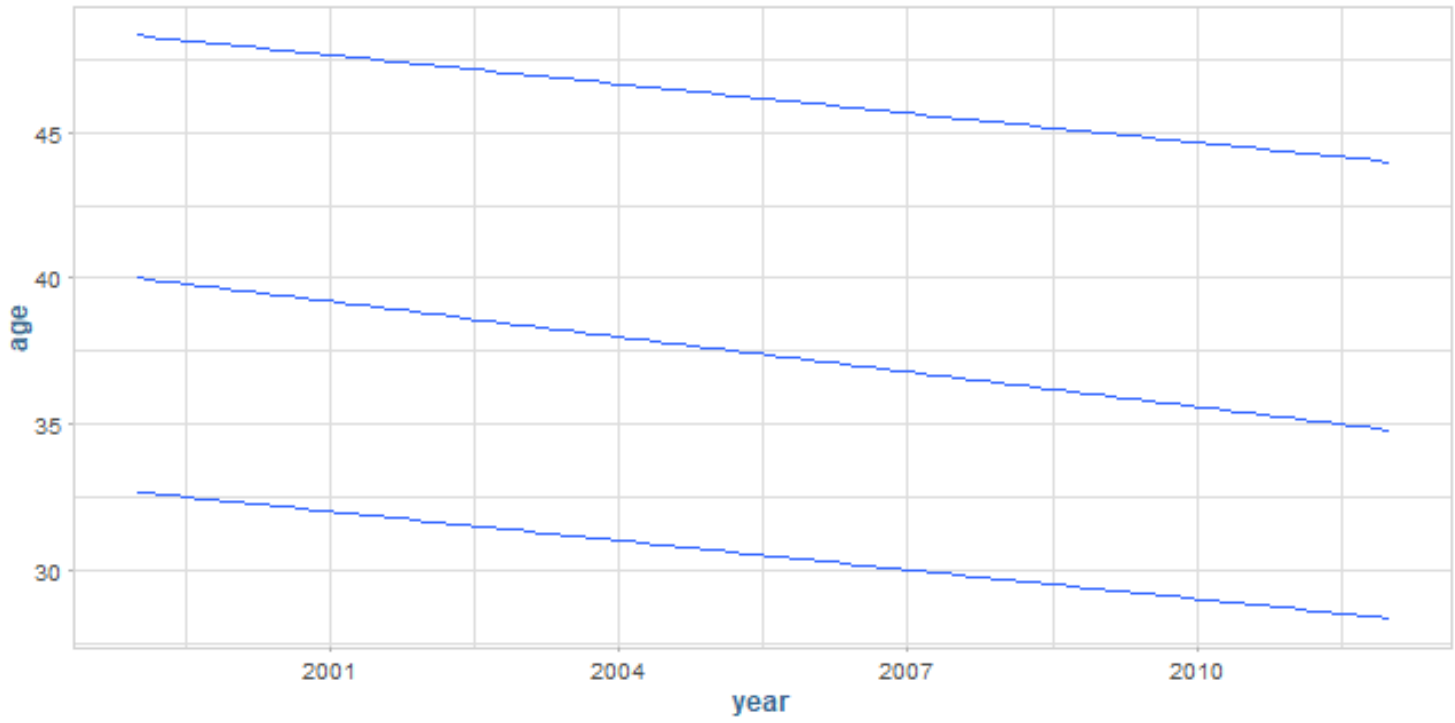
```
Warning: Removed 23 rows containing non-finite values (stat_boxplot).
```




```
ggplot(cbMen, aes(year, age)) +
   geom_quantile()
```

```
Warning: Removed 23 rows containing non-finite values (stat_quantile).
```

```
Smoothing formula not specified. Using: y ~ x
```



11.)

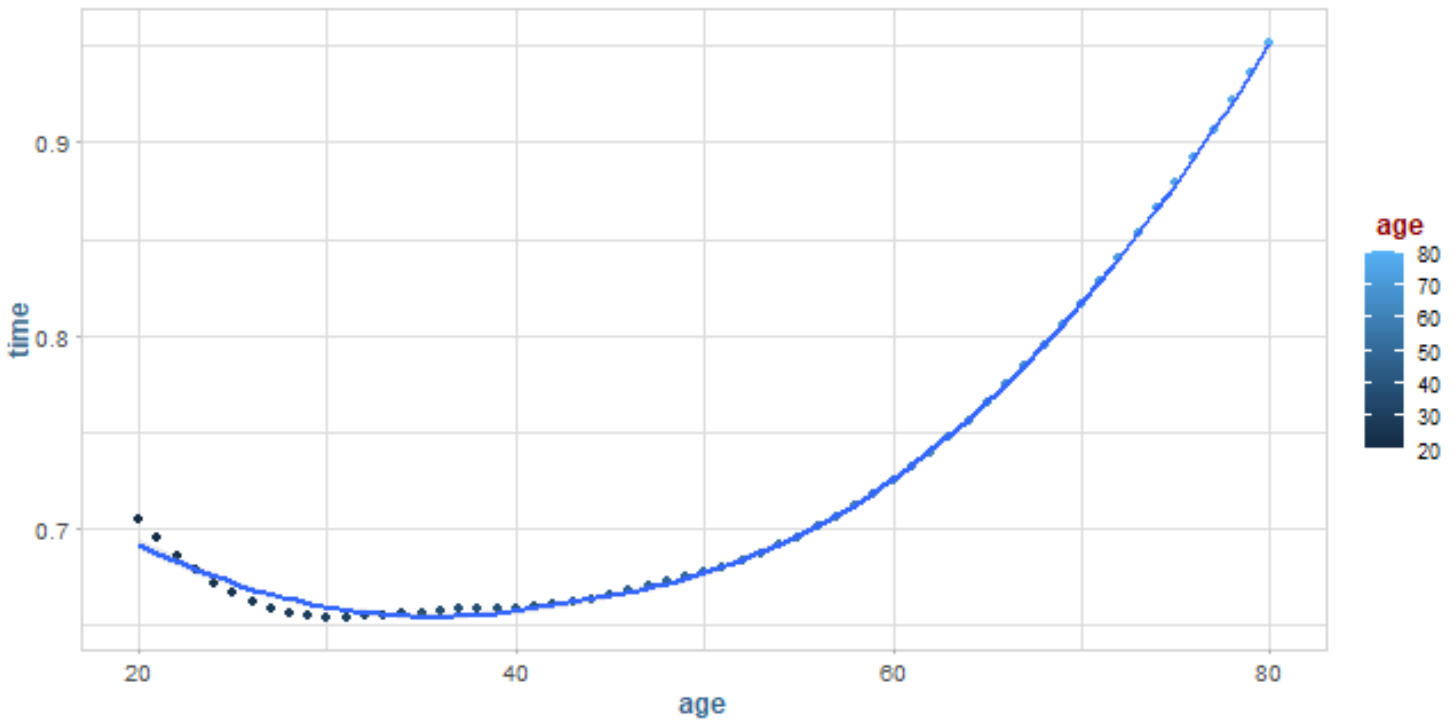
Normalize each male runner's time by the fastest time for the runner of the same age. To do this, find the fastest runnerk for each year of age from 20 to 80, tapply() function maybe helpful here. Smooth these times using loess() and find the smoothed fime using predict. Use these smoothed times to normalize each run time.

```
cbMenSub <- as.data.table(cbMenSub)

fastest_times <- cbMenSub[, .(max = max(runTime)), by = list(year, age)]

normalized_times <- cbMenSub[, .(runTime), by = list(year, age)]
normalized_times <- merge(normalized_times, fastest_times, by = c("year", "age"))
normalized_times[, normTime := runTime/max]

mR.norm <- loess(normTime ~ age, data = normalized_times)

summary(mR.norm)
```

```
Call:
loess(formula = normTime ~ age, data = normalized_times)

Number of Observations: 69735
Equivalent Number of Parameters: 5.11
Residual Standard Error: 0.1185
Trace of smoother matrix: 5.58  (exact)

Control settings:
  span     :  0.75
  degree   :  2
  family   :  gaussian
  surface  :  interpolate     cell = 0.2
  normalize:  TRUE
 parametric:  FALSE
drop.square:  FALSE
```

```
age_range <- seq( from = 20, to = 80, by = 1 )

smoothed_times <- predict(mR.norm, newdata = age_range)

smooth_results <- data.table(age = age_range, time = smoothed_times)

ggplot(smooth_results, aes(age, time)) +
   geom_point(aes(col = age)) +
   geom_smooth(alpha = .2)
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



12.)

Clean the strings in home and menRes to remove all leading and trailing blanks and multiple contiguous blanks. Also, make all letters lower case and remove any punctuation such as '.' or ','.

```
homeClean2 <- str_remove_all(homeClean, " ")
homeClean2 <- str_remove_all(homeClean2, ",")

head(homeClean2)
```

```
[1] "ethiopia" "kenya"    "kenya"    "kenya"    "kenya"    "kenya"
```

13.)

In section 2.5 we created an id for a runner by pasting together name, year of birth, and state. Consider using the home town instead of the state. How many runners have competed in at least 8 races using this new id?

```
cbMenSub2 <- cbMenSub

cbMenSub2$ID = paste(nameClean, homeClean2, cbMenSub$yob, sep = "_")

races <- tapply(cbMenSub2$year, cbMenSub2$ID, length)

races8 <- names(races)[which(races >= 8)]
```

```
men8 <- cbMenSub2[ cbMenSub2$ID %in% races8, ]

orderByRunner <- order(men8$ID, men8$year)

men8 <- men8[orderByRunner, ]
```

14.)

Further refine the set of athletes in the longitudinal analysis by dropping those IDs who have a large jump in time in consecutive races and who did not compete for two or more years in a row. How many unique IDs do you have when you include these additional restrictions?

```
numRaces <- tapply(cbMenSub$year, cbMenSub$ID, length)
races8 <- names(numRaces)[which(numRaces >= 8)]
men8 <- cbMenSub[ cbMenSub$ID %in% races8, ]

by_year <- as.data.table(men8)[, .(year, runTime), by = ID]

sans_outliers <- by_year %>%
   group_by(ID) %>%
   mutate(year_gap = year - lag(year, 1), run_diff = runTime - lag(runTime, 1)) %>%
   filter(year_gap <= 2) %>%
   filter(abs(run_diff) < 5) %>%
   distinct(ID)

nrow(sans_outliers)
```

```
[1] 305
```