Marathon Data Analysis

Hindrek Teder, Dmitri Timašjov, Ragnar Vent

Tuesday, May 26, 2015

Configuration

Libraries used in the project:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(rpart)
library(rpart.plot)
library(fpc)
library(ggplot2)
library(pheatmap)
library(scales)
```

Preprocessing

Data: SEB 17th Tartu Rattamaraton

The data was received from the official Club Tartu Maraton home page (https://tartumaraton.ee/en/results/) and included official results of SEB 17th Tartu Rattamaraton (89/40 km). The analysis of this project are based on the longer distance of the marathon (89 km). For the record, this pipeline can be used to analyse other Tartu maraton events.

```
#Read in raw data
data = read.csv2("data/rm_2014_lp.csv", header=T, skip = 5, na.strings="")
```

```
str(data)
```

```
## 'data.frame':    3065 obs. of  15 variables:
##  $ place        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ L.place      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ s.nr         : int  75 77 2 1 6 60 26 8 15 14 ...
```

```
##  $ name         : Factor w/ 3049 levels "Aagver Maris",..: 314 2385 1349 1997 151 2296 2831 2531 258
##  $ country      : Factor w/ 27 levels "Harju","Hiina",..: 10 10 22 22 23 15 26 22 16 22 ...
##  $ split.1      : Factor w/ 1424 levels "0:20:24","0:20:27",..: 2 1 5 9 6 9 8 6 7 3 ...
##  $ split.2      : Factor w/ 1500 levels "0:37:50","0:37:55",..: 2 4 4 3 5 5 4 5 4 1 ...
##  $ split.3      : Factor w/ 1717 levels "1:00:16","1:00:17",..: 2 3 8 7 8 5 8 5 4 2 ...
##  $ split.4      : Factor w/ 2212 levels "1:25:17","1:25:18",..: 1 2 8 6 7 6 5 9 4 1 ...
##  $ split.5      : Factor w/ 2266 levels "1:53:02","1:53:03",..: 3 1 9 4 8 4 5 9 8 2 ...
##  $ split.6      : Factor w/ 1994 levels "2:11:02","2:11:03",..: 1 2 5 5 6 5 7 6 7 3 ...
##  $ time         : Factor w/ 2388 levels "2:29:11","2:29:12",..: 1 2 3 3 3 3 4 5 5 6 ...
##  $ age.group    : Factor w/ 21 levels "M17","M20","M21",..: 3 3 6 4 3 3 2 4 3 2 ...
##  $ place2       : int  1 2 1 1 3 4 1 2 5 2 ...
##  $ particip.time: int  2 3 16 7 16 4 9 15 9 7 ...
```

Features of the data:

- place - overall ranking
- L.place - female ranking
- s.nr - starting number
- name - name of the competitor
- country - county of resident for Estonian, country of resident for foreigner
- split.1 - time in Matu (12.3 km)
- split.2 - time in Ande (22.9 km)
- split.3 - time in Puka (36.5 km)
- split.4 - time in Astuvere (50.6 km)
- split.5 - time in Palu (66.3 km)
- split.6 - time in Hellenurme (77.2 km)
- time - finishing time (89.0 km)
- age.group - groups by gender and age
- place2 - age.group ranking
- particip.time - who many times have participated before (including this time)

Functions

For easier comparison we converted split and time strings to the base unit of a second.

```r
#Function to convert time string to seconds
charToSec = function(x){
  if(!is.na(x)){
    incr = c(3600, 60, 1)
    vals = sapply(strsplit(as.character(x),":"), FUN=function(y){as.numeric(y)})
    return(sum(incr*vals))
  }else{
    return(NA)
  }
}
```

```r
#Convert timestamps to seconds
for(i in 1:6){
  data[, paste("split.",i,sep="")]=sapply(data[,paste("split.",i,sep="")], FUN=function(x){charToSec(x)
}
data[, "time"]=sapply(data[, "time"], FUN=function(x){charToSec(x)})
```

Imputation

The raw data have in total of 3065 objects and only 8 of them were incomplete. As this is less than 0.3 percentange from the whole there is no need for the data imputation.

```
#Data imputation (currently we just leave rows with missing data out)
data = data[rowSums(is.na(data[,paste("split.",1:6,sep="")]))==0,]
```

Added features

```
#Add gender
data$gender <- 0
data[is.na(data[,"L.place"]),]$gender <- "male"
data[!is.na(data[,"L.place"]),]$gender <- "female"

#Add unisex agegroup
data$age.group2 <- as.numeric(substr(data$age.group, 2, 3))

#Add nationality
countries <- levels(data$country)
counties <- countries[c(1,3,4,7,8,11,12,14,15,16,18,22,23,24,26,27)]
data$nationality <- 0
data[is.element(data$country, counties),]$nationality <- "Estonia"
data[!is.element(data$country, counties),]$nationality <- "foreign"

#Add county
data$county <- data$country
levels(data$county) <- c(levels(data$county), "other")
data$county[!is.element(data$country, counties)] <- "other"
```

```
#Split final times into 10 groups
data$timeCategory <- ntile(data$time, 10)

#Split start numbers into 10 groups
data$sNrCategory <- ntile(data$s.nr, 5)

#Split number of participations into 10 groups
data$participTimeCategory <- ntile(data$particip.time, 5)

#Split final place into 10 groups
data$placeCategory <- ntile(data$place, 10)
```

```
#Combine all Estonian participants
data$countryCategory <- data$country
levels(data$countryCategory) <- c(levels(data$countryCategory), "Eesti")
data[data$country %in% c(
  "Harju", "Hiiumaa", "Ida-Viru", "Jõgeva", "Järvamaa", "Lääne-Viru", "Läänemaa",
  "Pärnu", "Rapla", "Saaremaa", "Tallinn", "Tartu", "Valga", "Viljandi", "Võru", "Põlva"), "countryCate
```

```r
#Combine age categories
data$ageCategory <- data$age.group2
levels(data$ageCategory) <- c(levels(data$ageCategory), c("17-21", "35-45", "50-60", "65+"))
data[data$age.group2 %in% c("17", "20", "21"), "ageCategory"] <- "17-21"
data[data$age.group2 %in% c("35", "40", "45"), "ageCategory"] <- "35-45"
data[data$age.group2 %in% c("50", "55", "60"), "ageCategory"] <- "50-60"
data[data$age.group2 %in% c("65", "70", "75"), "ageCategory"] <- "65+"
```

Fix preprocessed data

```r
#Write out preprocessed data
write.table(data, "data/processedData.txt", sep="\t", row.names=F)

#Write out split distances
dist = data.frame(0.0, 12.3, 22.9, 36.5, 50.6, 66.3, 77.2, 89.0)
splitNames <- c("Start", "Matu", "Ande", "Puka", "Astuvere", "Palu", "Hellenurme", "Finish")
colnames(dist) <- splitNames
write.table(dist, "data/distances.txt", sep="\t", row.names=F)

#Calculate and write out distances between splits
splits = c(0)
for(i in 2:8){
  splits <-  c(splits, dist[i] - dist[i-1])
}
splits <- as.data.frame(splits)
colnames(splits) <- splitNames
write.table(splits, "data/splits.txt", sep="\t", row.names=F)
```

Additional data: dist

This data combines three different data sets:

- SEB 17th Tartu Rattamaraton
- Road Administration (http://www.mnt.ee/kaugus/m/)
- Municipality portal (http://portaal.ell.ee/)

```r
#Distance and population data
distance <- c(186,305,130,53,103,123,249,49,174,157,328,186,0,86,78,71)
participants <- c(434,6,40,43,44,103,35,54,88,56,36,755,605,114,67,71)
population <- c(153648,9709,153312,32275,31688,61099,25513,29169,85539,34989,34485,434339,148673,31790,4
dist <- data.frame(counties, population, distance, participants)
```

```r
str(dist)
```

```
## 'data.frame':    16 obs. of  4 variables:
##  $ counties    : Factor w/ 16 levels "Harju","Hiiumaa",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ population  : num  153648 9709 153312 32275 31688 ...
##  $ distance    : num  186 305 130 53 103 123 249 49 174 157 ...
##  $ participants: num  434 6 40 43 44 103 35 54 88 56 ...
```

Features of the data:

- counties - county name
- population - population of the county
- distance - county seat distance from Tartu
- participants - total number of participants from the county

Additional data: winningResultsByYear

The data includes winning times of 7 SEB Tartu Rattamaraton competitions.

```
#Read in data
history = read.table("data/winningResultsByYear.txt", header=T)
```

```
str(history)
```

```
## 'data.frame':    7 obs. of  2 variables:
##  $ Year: int  2008 2009 2010 2011 2012 2013 2014
##  $ Time: Factor w/ 7 levels "2:29:11","2:30:47",..: 6 4 7 2 5 3 1
```

Features of data:

- Year - year of the competition
- Time - best finish time

Descriptive Statistics

```
#Read in data
data <- read.table("data/processedData.txt", header=T)
```

Statistical analysis

t-test

Determine if two sets of data are significantly different from each other.

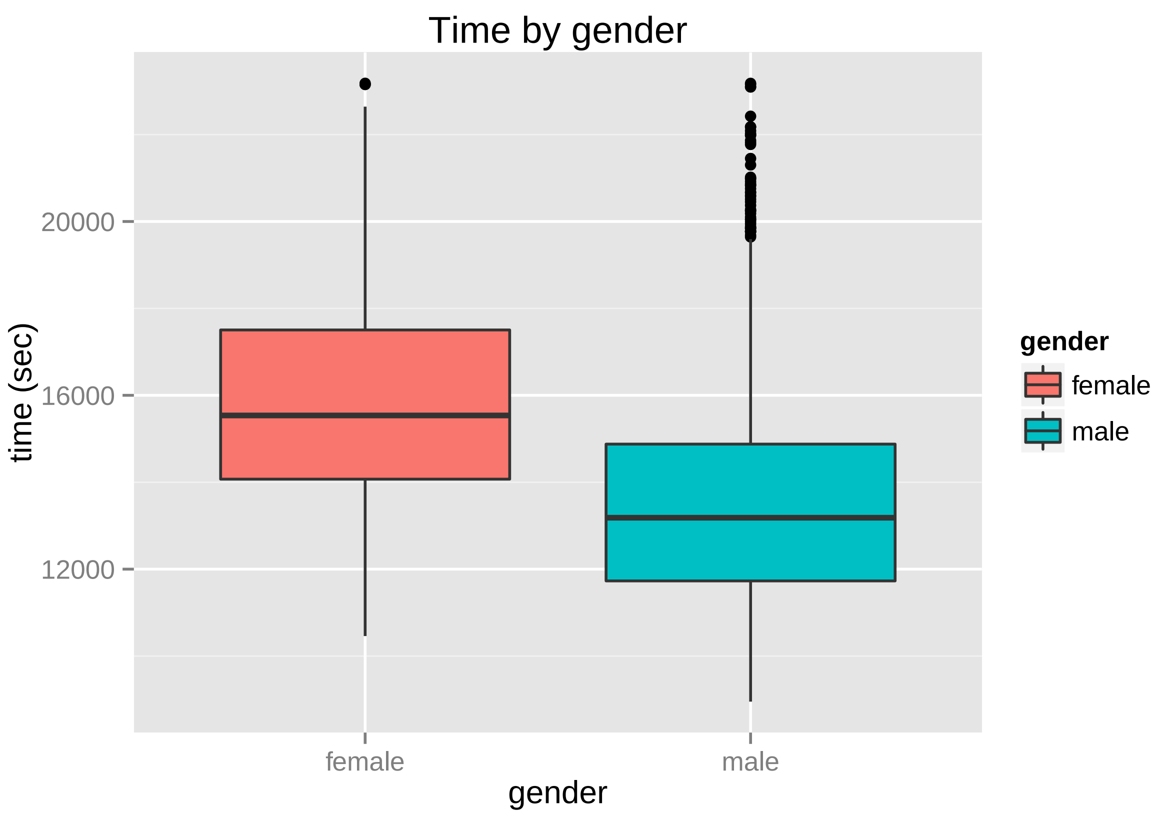
```
t.test(data$time~data$gender)
```

Time by gender

```
##
##  Welch Two Sample t-test
##
## data:  data$time by data$gender
```

```
## t = 11.772, df = 222.39, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2019.255 2831.273
## sample estimates:
## mean in group female   mean in group male
##             15882.71             13457.45
```

```
ggplot(data, aes(x = gender, y = time, fill = gender)) +
  geom_boxplot() +
  labs(title = "Time by gender", y = "time (sec)")
```



Result: finishing time is significantly different between genders.

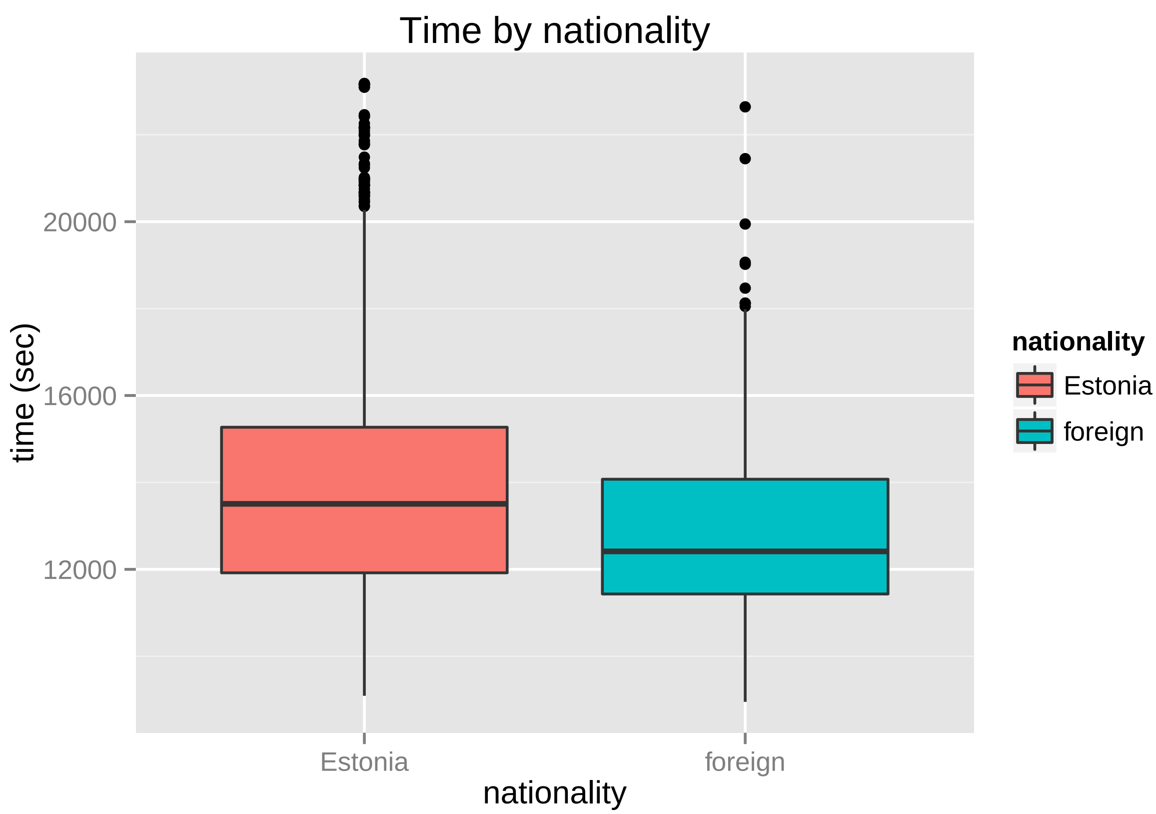
```
t.test(data$time~data$nationality)
```

Time by nationality

```
##
##  Welch Two Sample t-test
##
## data:  data$time by data$nationality
## t = 9.0438, df = 818.08, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   740.1566 1150.5074
## sample estimates:
## mean in group Estonia mean in group foreign
##              13775.76              12830.43
```

```
ggplot(data, aes(x = nationality, y = time, fill = nationality)) +
  geom_boxplot() +
  labs(title = "Time by nationality", y = "time (sec)")
```



Result: finishing time is significantly different between Estonians and foreigners.

ANOVA

Determine if two or more sets of data are significantly different from each other.

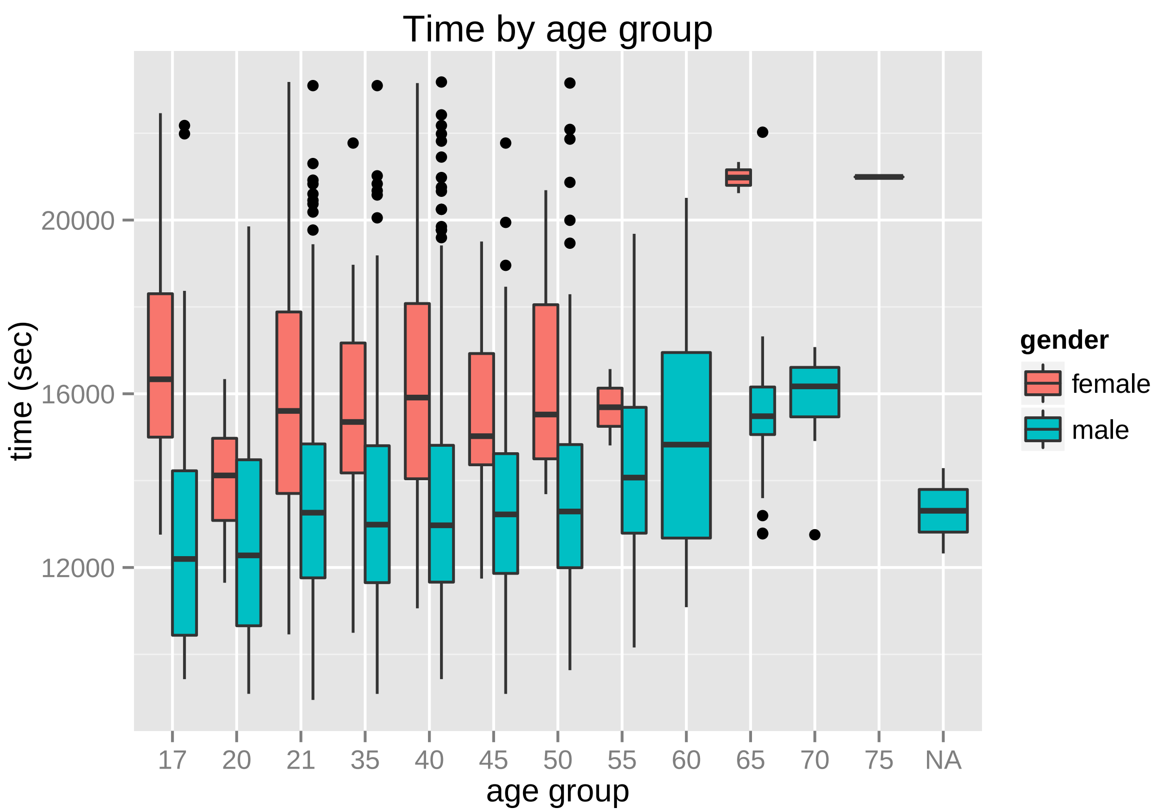
```
summary(aov(data$time~data$age.group2))
```

Time by age group

```
##                    Df    Sum Sq  Mean Sq F value   Pr(>F)
```

```
## data$age.group2    1 7.769e+07 77694805   13.07 0.000304 ***
## Residuals       3053 1.814e+10  5942932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness
```

```
ggplot(data, aes(x = factor(age.group2), y = time, fill = gender)) +
  geom_boxplot() +
  labs(title = "Time by age group", x = "age group", y = "time (sec)")
```



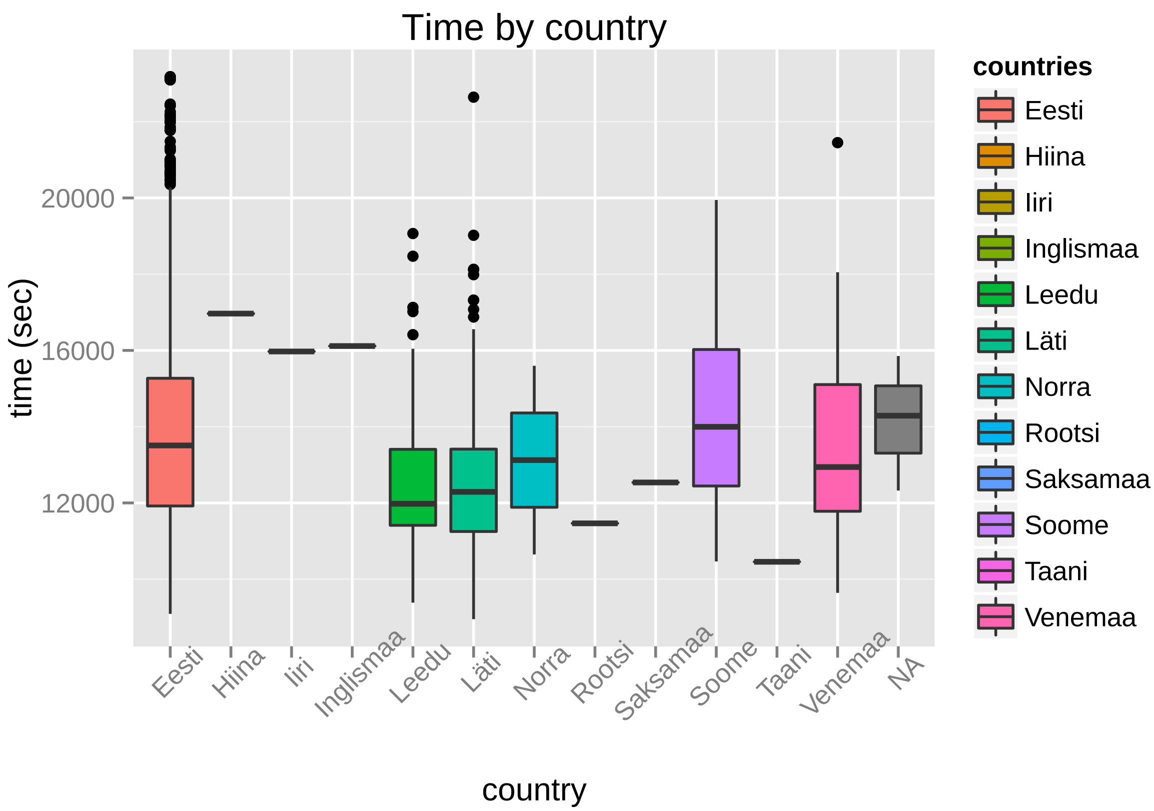
Result: finishing time is significantly different between age groups.

```
summary(aov(data$time~data$country))
```

Time by country

```
##                 Df    Sum Sq  Mean Sq F value   Pr(>F)
## data$country   26 7.043e+08 27090028   4.683 5.29e-14 ***
## Residuals    3027 1.751e+10  5785309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness
```

```r
ggplot(data, aes(x = factor(countryCategory), y = time, fill = countryCategory)) +
  geom_boxplot() +
  labs(title = "Time by country", x = "country", y = "time (sec)", fill = "countries") +
  theme(axis.text.x = element_text(angle = 45))
```



Result: finishing time is significantly different between countries.

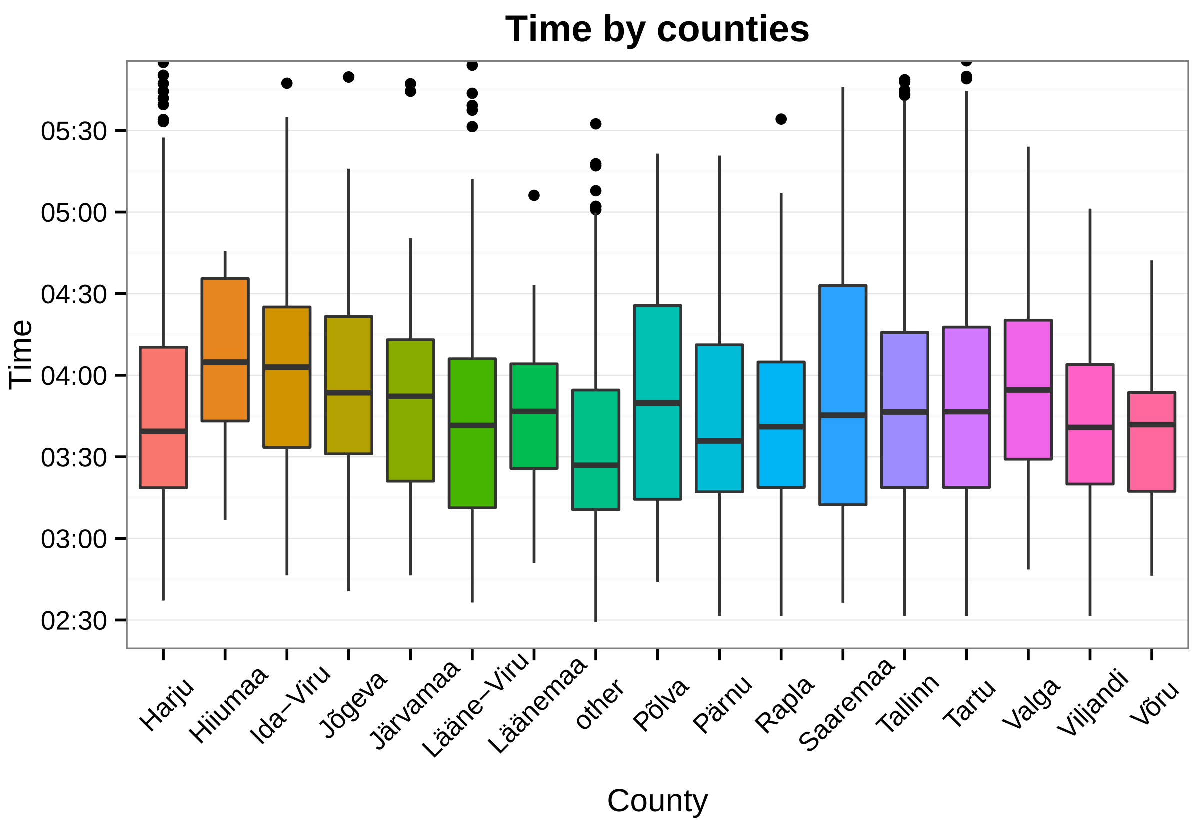
```r
summary(aov(data$time~data$county))
```

Time by county

```
##                  Df    Sum Sq  Mean Sq F value   Pr(>F)
## data$county      16 5.416e+08 33847049   5.819 1.11e-12 ***
## Residuals      3040 1.768e+10  5816460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
ggplot(data, aes(x = county, y = as.POSIXct(time, tz = "GMT", origin = "2014-09-21"), fill = county)) +
  geom_boxplot() +
  labs(title = "Time by counties", x = "County", y = "Time") +
  theme_bw() +
  theme(panel.grid.major.x=element_blank(),
```

```
        plot.title = element_text(lineheight=.8, face="bold", vjust=1),
        axis.text.x=element_text(angle=45, vjust = 0.7),
        legend.position="none") +
  scale_y_datetime(breaks=date_breaks("30 min"), labels=date_format("%H:%M"))
```



Result: finishing time is significantly different between counties.

Chi-square test

Determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories.

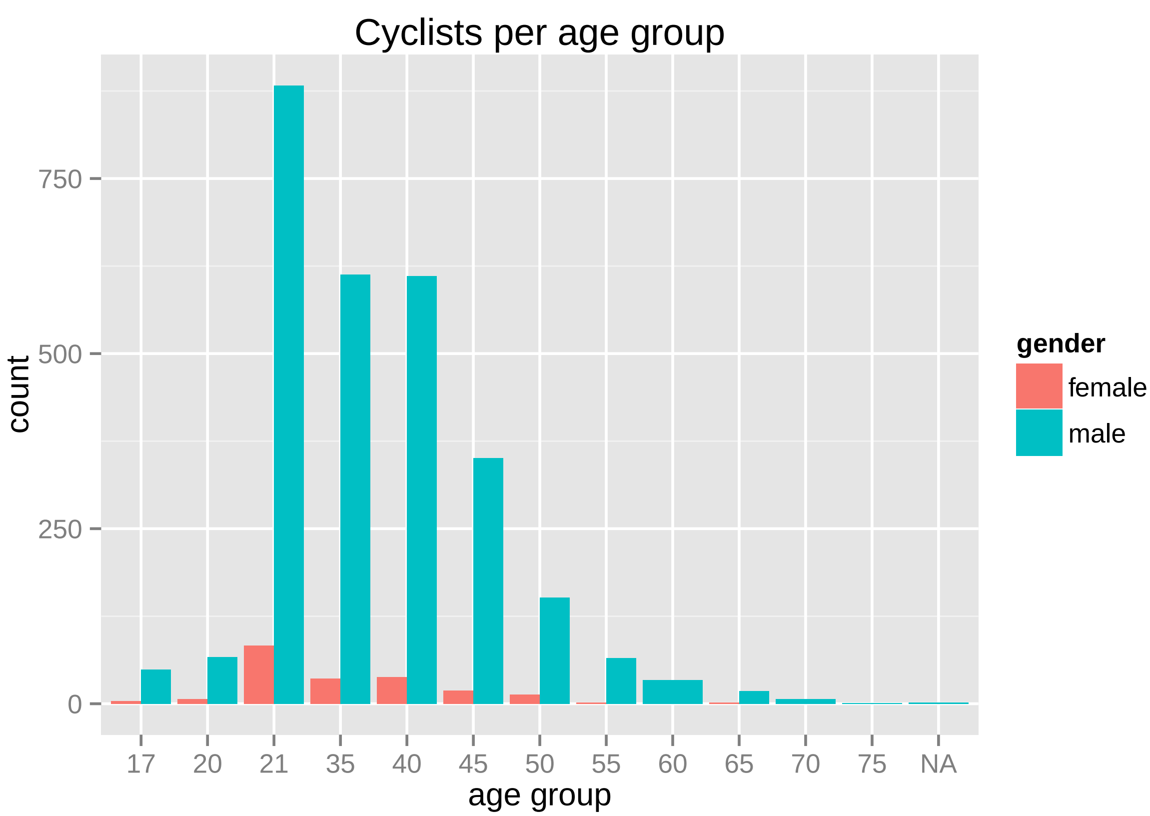
Some of the age groups have to be combined to fulfil the assumption of Chi-square test - frequency of every group need to be at least 5.

```
tbl <- table(data$gender, data$age.group2)
ctbl <- cbind(tbl[,"17"]+tbl[,"20"],
                tbl[,"21"],
                tbl[,"35"],
                tbl[,"40"],
                tbl[,"45"],
                tbl[,"50"]+tbl[,"55"]+tbl[,"60"]+tbl[,"65"]+tbl[,"70"]+tbl[,"75"])
colnames(ctbl) = c("[15-21)","[21-22)","[22-36)","[36-41)","[41-46)","[46-76)")
```

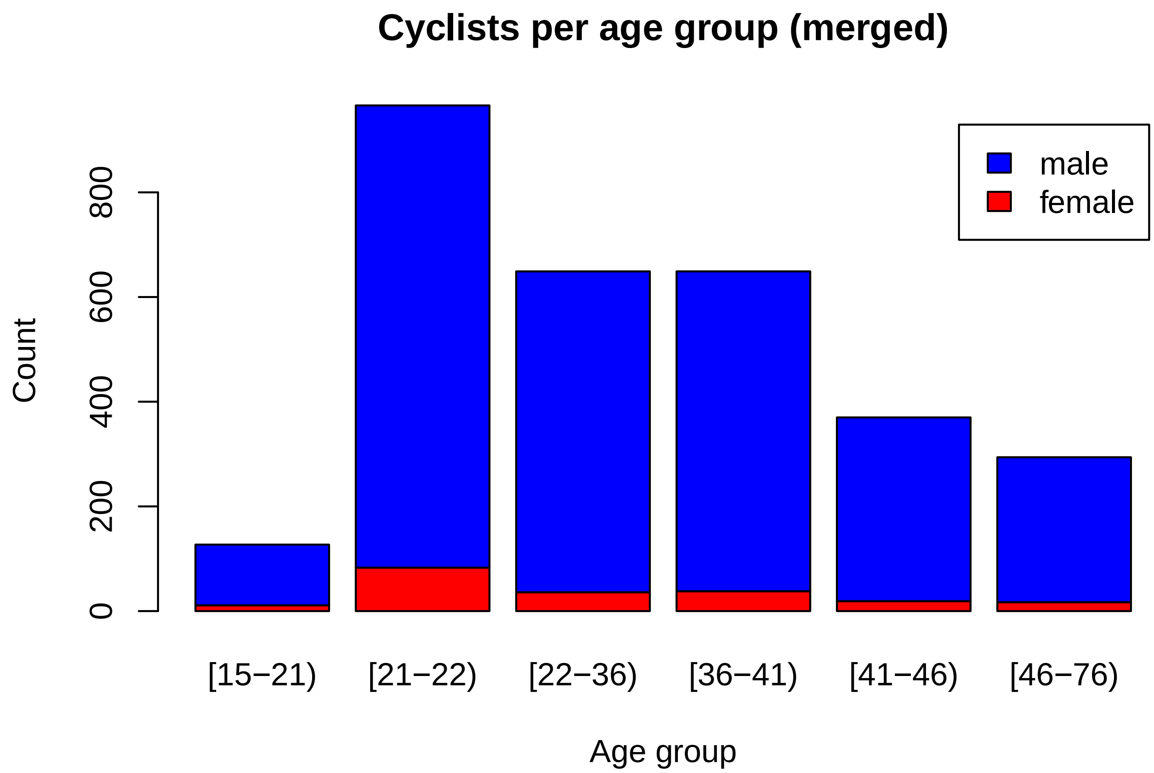
```r
chisq.test(ctbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ctbl
## X-squared = 10.31, df = 5, p-value = 0.0669
```

```r
ggplot(data, aes(x = factor(age.group2), fill = gender)) +
  geom_bar(position = "dodge") +
  labs(title = "Cyclists per age group", x = "age group")
```




```r
barplot(ctbl, col = c("red","blue"), legend = T,
        main = "Cyclists per age group (merged)",
        xlab = "Age group",
        ylab = "Count")
```



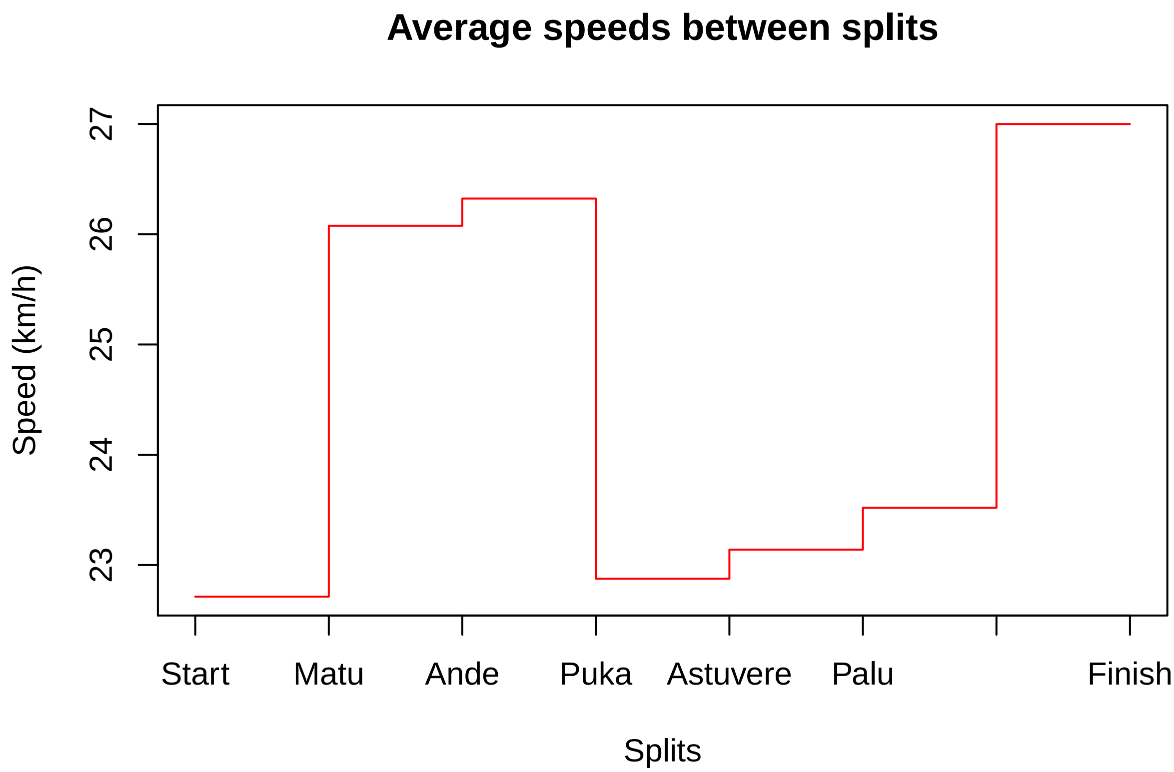
Result: no significant difference in frequency distribution of age groups between genders.

Additional plots

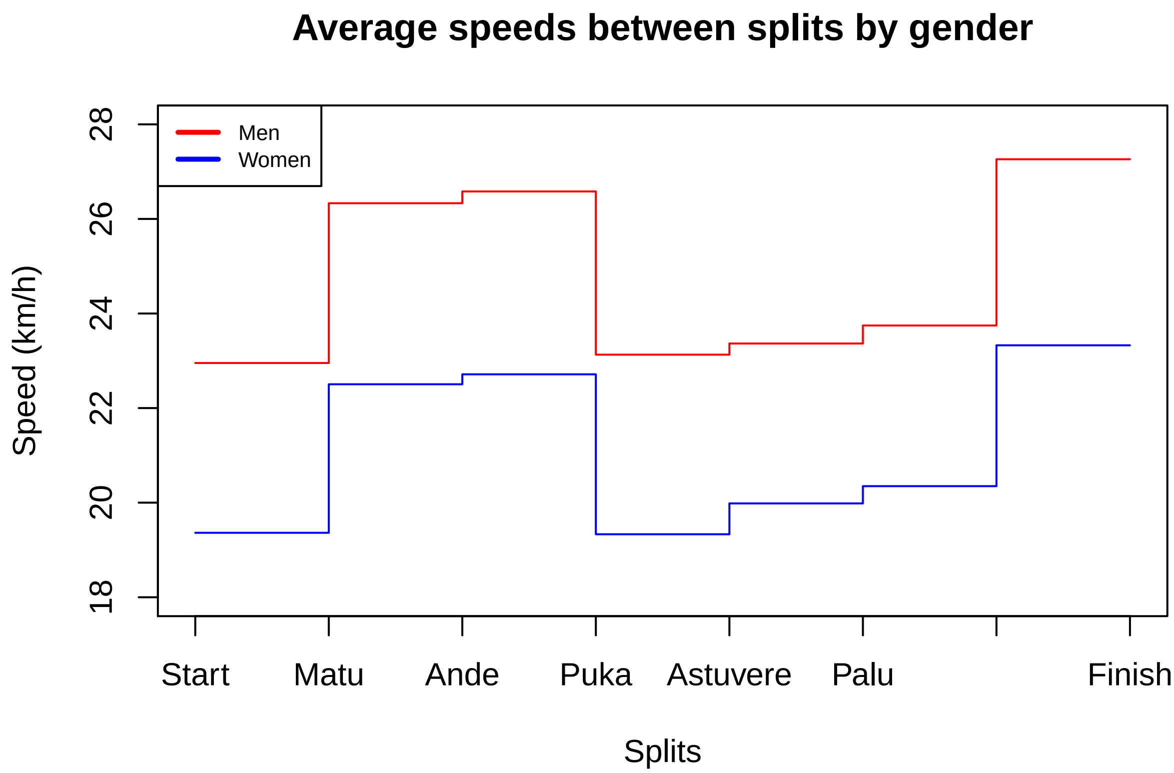
Speed between splits

```
splits <- read.table("data/splits.txt", header=T)

#Find speed based on given subset of data
findSpeeds = function(x, splits){
  speed = mean(splits[,1]  / (x[,1]/3600))
  speeds = c(speed, speed)
  for(i in 2:length(splits)){
    speed = mean(splits[,i] / ((x[,i] - x[,i-1])/3600))
    speeds = c(speeds, speed, speed)
  }
  return(speeds)
}

#Calculate speeds for each subset
overallSpeeds = findSpeeds(data[,c(paste("split.",1:6,sep=""),"time")], splits[-1])
menSpeeds = findSpeeds(data[data[,"gender"]=="male",c(paste("split.",1:6,sep=""),"time")], splits[-1])
womenSpeeds = findSpeeds(data[data[,"gender"]=="female",c(paste("split.",1:6,sep=""),"time")], splits[-
```

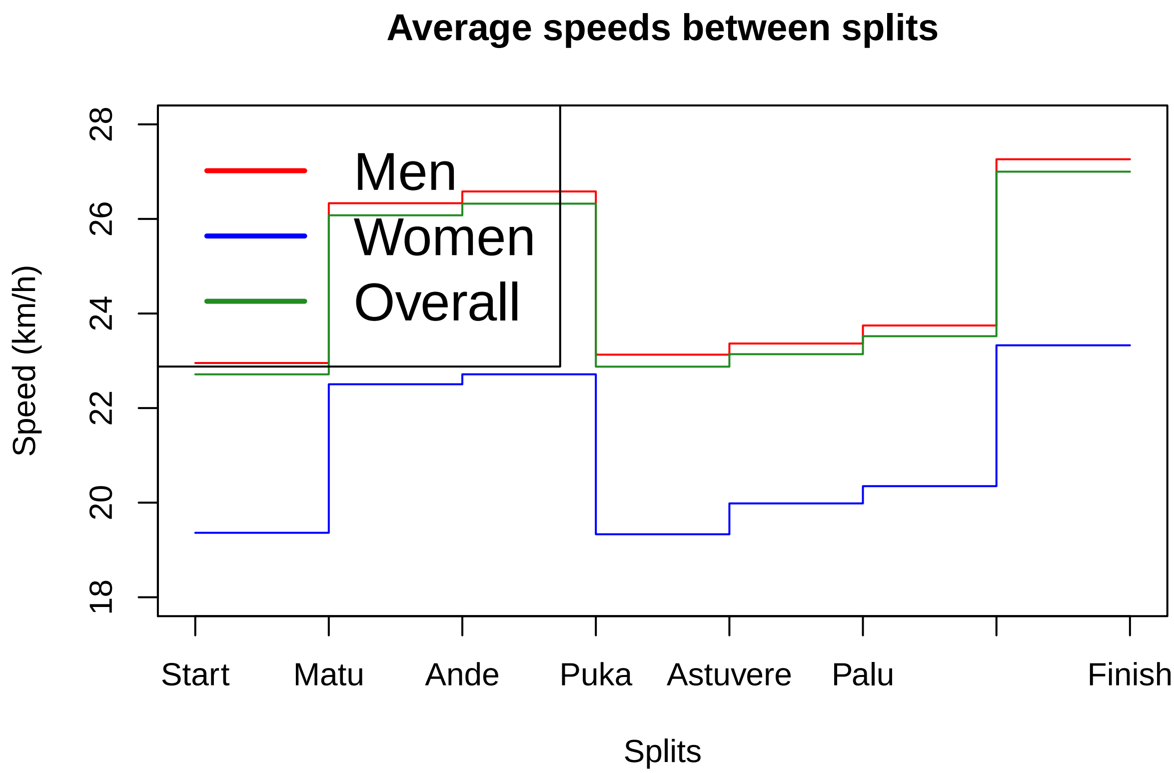
```r
#Draw plots
xValues = c(0,sort(rep(1:6, 2)),7)
plot(xValues, overallSpeeds, type = "l", col="red", xlab="Splits", ylab = "Speed (km/h)",
     xaxt="n", main="Average speeds between splits")
axis(1, at=0:7, labels= colnames(splits))
```




```r
#By gender
plot(xValues, menSpeeds, type = "l", col="red", xlab="Splits", ylab = "Speed (km/h)",
     xaxt="n", main="Average speeds between splits by gender", ylim=c(18,28))
lines(xValues, womenSpeeds, col="blue")
axis(1, at=0:7, labels= colnames(splits))
legend("topleft", legend = c("Men","Women"), lty=c(1,1), lwd=c(2.5,2.5), col=c("red","blue"), cex=.7)
```



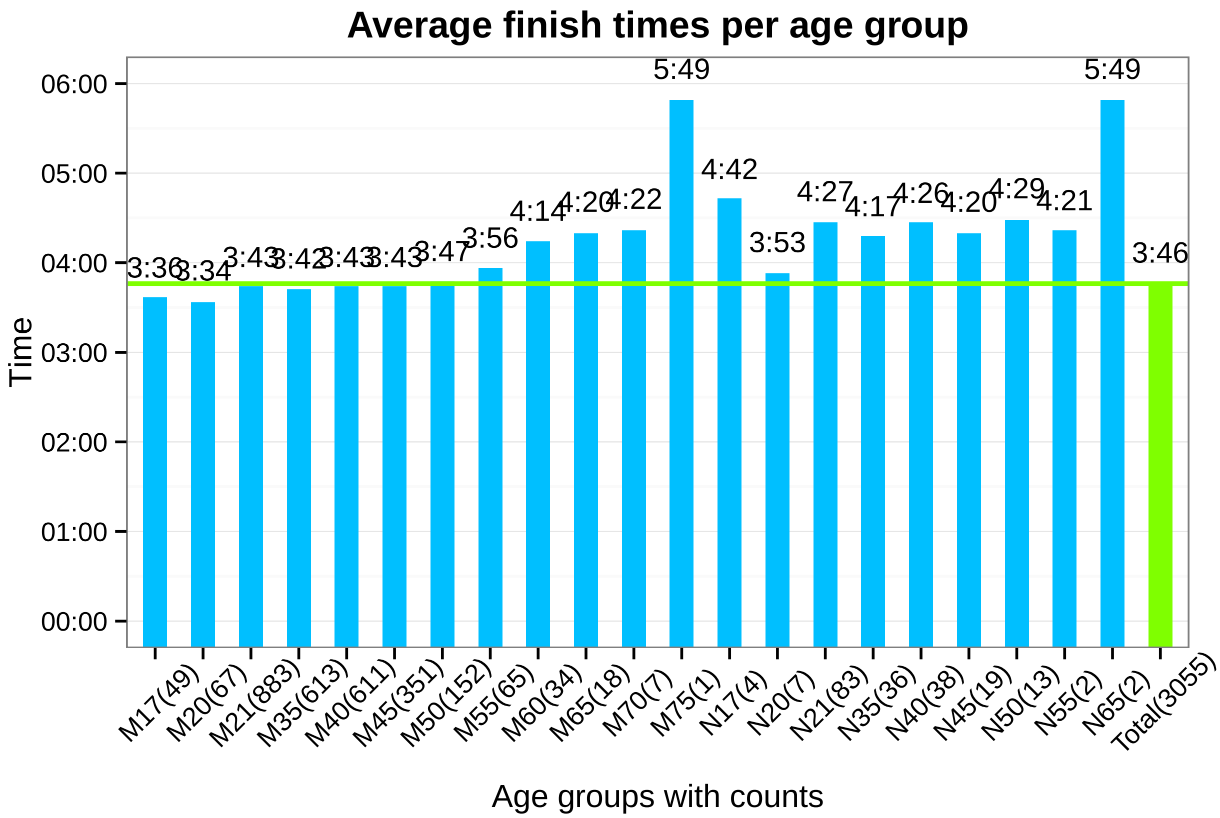

```
#All together
plot(xValues, menSpeeds, type = "l", col="red", xlab="Splits", ylab = "Speed (km/h)",
     xaxt="n", main="Average speeds between splits", ylim=c(18,28))
lines(xValues, womenSpeeds, col="blue")
lines(xValues, overallSpeeds, col="forestgreen")
axis(1, at=0:7, labels= colnames(splits))
legend("topleft", legend = c("Men","Women","Overall"), lty=c(1,1,1),
       lwd=c(2.5,2.5,2.5), col=c("red","blue","forestgreen"), cex=1.7)
```



Average finish times per age group

```
#Average finish times per age group
meanClass = function(data, class){
  if(class=="Total"){
    res = mean(data[, "time"], na.rm=T)
  }else{
    res = mean(data[data[,"age.group"]==class, "time"], na.rm=T)
  }
  hours = floor(res / 3600)
  minutes = floor((res - (3600*hours))/60)
  return(paste(hours,":",minutes,sep=""))
}

tab = table(data$age.group)
x = c(paste(names(tab),"(" ,tab, ")",sep=""), paste("Total(",sum(tab),")", sep=""))
y = as.POSIXct(sapply(c(names(tab),"Total"), FUN = function(x){meanClass(data, x)}), format="%H:%M")
xy=data.frame(x, y)


ggplot(xy, aes(x=xy$x, y=xy$y, width=0.5)) +
  geom_bar(stat="identity",
           fill=c(rep("deepskyblue",length(xy$y)-1), "chartreuse"))+
  geom_text(aes(label=substr(xy$y,13,16)), vjust=-1, size=4) +
```

```r
  xlab("Age groups with counts") + ylab("Time") +
  ggtitle("Average finish times per age group")+
  theme_bw()+
  theme(panel.grid.major.x=element_blank(),
        plot.title = element_text(lineheight=.8, face="bold", vjust=1),
        axis.text.x=element_text(angle=45, vjust = 0.7))+
  scale_y_datetime(limits=c(as.POSIXct('0:00',format="%H:%M"),
                            as.POSIXct('6:00',format="%H:%M")))+
  geom_hline(aes(yintercept = as.numeric(y[length(y)])), colour = "chartreuse",size=0.8)
```

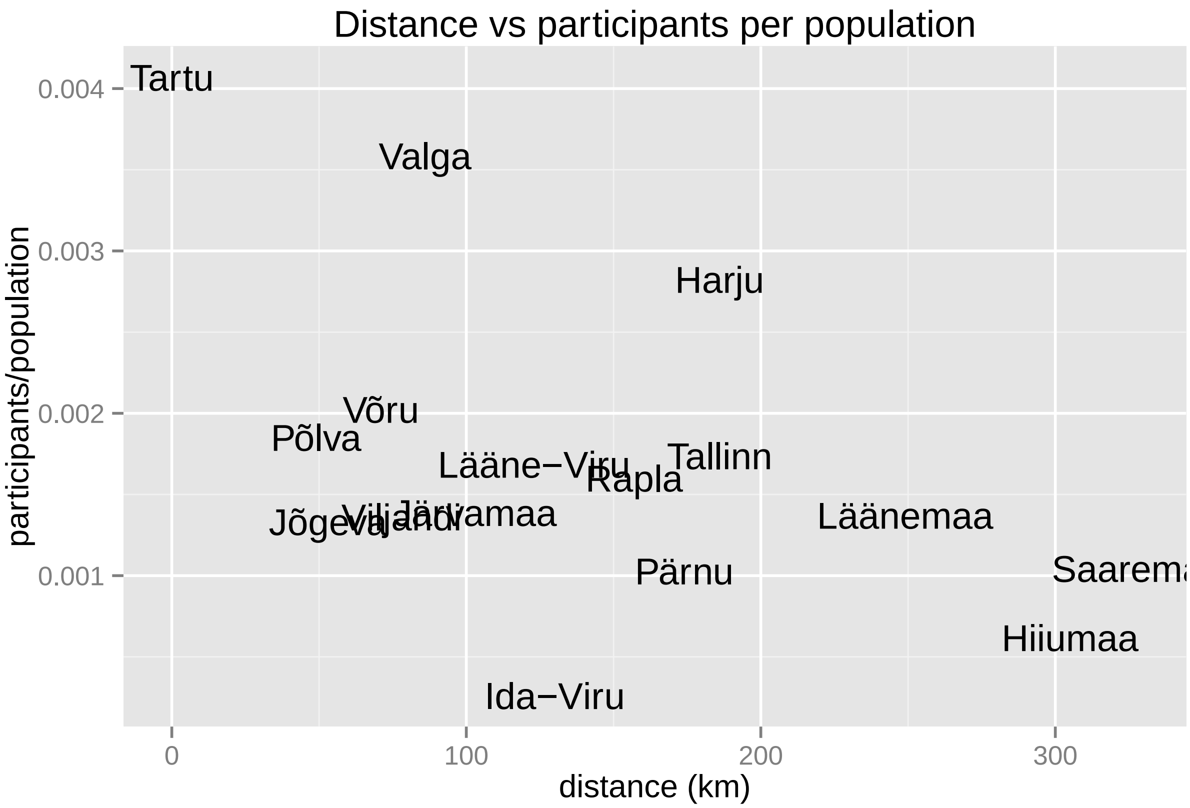


Distance vs participants per population

```r
#Distance vs participants per population
cor(dist$distance, dist$participants/dist$population, use = "complete.obs", method = "kendall")
```

```
## [1] -0.3096261
```

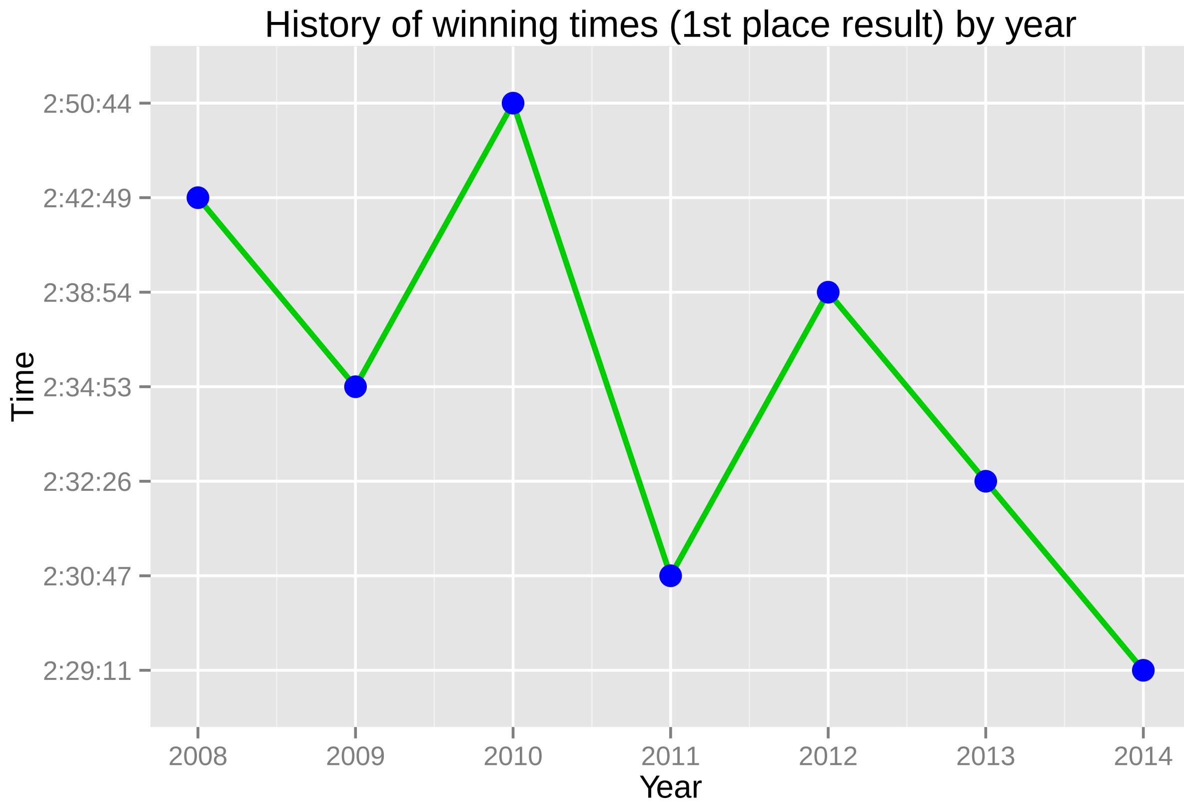
```r
ggplot(dist, aes(x = distance, y = participants/population, label = counties)) +
  geom_text() +
  labs(title = "Distance vs participants per population", x = "distance (km)")
```



Result: medium negative correlation between distance from Tartu and participants per population.

Winning time

```
ggplot(history, aes(x=Year, y=Time, group=1)) +
  geom_line(size=1, colour="green3") +
  geom_point(size=4, colour="blue") +
  ggtitle("History of winning times (1st place result) by year") +
  scale_x_continuous(breaks=c(2008:2014), labels=c(2008:2014))
```



Clustering

```r
#Read in data
data = read.table("data/processedData.txt", header=T)
```

DBSCAN

A density-based clustering algorithm was used in this project. DBSCAN groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). This principle fits with the project nature.

Currently we find people who were in the same pace group throughout the entire race (at least 2 people to make a pace group).

```r
ds = dbscan(data[,c(paste("split.",1:6,sep=""), "time")], MinPts=2, eps=10)

#Order data by cluster
dsData = cbind(data, ds$cluster)
dsData = dsData[order(ds$cluster),]

#Show clusters
dsData[dsData[,"ds$cluster"] > 0, -c(1:3,5,13:24),][c(1,8,10)]
```

```
##                      name  time ds$cluster
## 3        Maasikmets Alges  9090          1
## 4             Pütsep Erki  9090          1
## 5            Austa Caspar  9090          1
## 6          Schultz Silver  9090          1
## 7           Vaidem Josten  9091          1
## 8          Tamkõrv Helmet  9092          1
## 12            Kriit Kalle  9093          1
## 13    Ottender Sten-Erik  9148          2
## 15             Loo Martin  9155          2
## 22           Valvas Vahur  9427          3
## 23         Oolo Kristjan  9428          3
## 24          Kiskonen Siim  9429          3
## 26          Kivistik Gert  9429          3
## 27         Manikas Domas  9430          3
## 29      Sertvytis Donatas  9430          3
## 31             Pallo Rait  9430          3
## 32           Kattai Kaupo  9430          3
## 33             Palm Tõnno  9430          3
## 34            Veski Tanel  9431          3
## 35    Pacevicius Šarunas  9431          3
## 36           Põldma Mirko  9432          3
## 37          Neemela Tarmo  9432          3
## 38              Olle Raul  9432          3
## 28            Öpik Oliver  9430          4
## 30       Kannimäe Viljar  9430          4
## 46          Stalberg Tair  9636          5
## 47       Kirsipuu Toomas  9636          5
## 48      Lauk Karl-Patrick  9637          5
## 49      Gristsenko Andrei  9637          5
## 52             Sala Raivo  9639          5
## 51             Nook Reimo  9637          6
## 53          Dzalbs Gunars  9641          6
## 54         Nikolaev Fedor  9641          6
## 59      Balgabaev Ravshan  9833          7
## 63           Pungar Urmas  9836          7
## 68             Post Margo  9836          7
## 70             Lehto Tiit  9837          7
## 71          Ridamäe Aivar  9838          7
## 60           Nõlvik Lasse  9833          8
## 61       Randma Kristjan  9834          8
## 65         Tõnisson Tiimo  9836          8
## 66       Kushnir Aleksei  9836          8
## 67           Rattur Rajko  9836          8
## 69         Malsroos Lauri  9836          8
## 72            Tuisk Priit  9842          8
## 76       Ivanov Vladimir  9976          9
## 100            Molev Juri  9981          9
## 77        Pelaitis Arnas  9977         10
## 80        Prangel Kristo  9978         10
## 91              Arak Anti  9980         10
## 99          Kirsipuu Tiit  9981         10
## 101          Välbe Urmas  9981         10
## 106          Lepik Toomas  9982         10
```

```
## 113          Teteris Janis  9985          10
## 117        Mavchun Georgii  9988          10
## 81            Vähi Markus  9978          11
## 86           Ukins Valdis  9979          11
## 87            Parv Martin  9979          11
## 89        Flaksis Martins  9980          11
## 82          Kuljus Viljar  9978          12
## 84     Zdeblovskiy Alexey  9979          12
## 85           Kollo Andres  9979          12
## 107        Zimelis Aigars  9983          12
## 110        Strazdins Rego  9984          12
## 94           Lukin Vitalik  9980          13
## 102          Roskoss Janis  9982          13
## 95             Lipp Aivar  9980          14
## 108          Linnus Sander  9983          14
## 98            Sügis Harri  9981          15
## 103        Kallari Taimar  9982          15
## 136        Maarits Andres 10158          16
## 138       Kannimäe Mihkel 10159          16
## 166        Danilas Meelis 10257          17
## 167           Nael Margus 10257          17
## 168            Ott Indrek 10257          17
## 171            Kivi Margo 10258          17
## 173       Inovskis Nauris 10259          17
## 201      Andersons Ainars 10427          18
## 205        Külanurm Karli 10430          18
## 209   Grigorovitsh Jaanus 10433          18
## 220      Uibokand Janelle 10460          19
## 229         Kaljumäe Aivo 10464          19
## 227           Kruus Kaupo 10463          20
## 230          Künnap Janis 10465          20
## 228          Suluste Jüri 10463          21
## 231             Rahi Tõnu 10471          21
## 237      Birkants Roberts 10533          22
## 242          Vevers Girts 10535          22
## 244         Padumäe Vaido 10537          23
## 249           Haava Henno 10540          23
## 289       Lejins Dzintars 10705          24
## 291              Hio Siim 10706          24
## 397           Jaaska Timo 11060          25
## 402         Losins Guntis 11062          25
## 593       Gavelis Povilas 11494          26
## 594  Tarabrinas Liutauras 11494          26
## 644            Levans Ivo 11616          27
## 645        Freinats Gints 11617          27
## 1130         Mooste Tarmo 12487          28
## 1131         Teepere Egon 12487          28
## 1424           Morel Ülar 13104          29
## 1427        Kannimäe Anne 13104          29
## 1691        Valgmäe Taavi 13668          30
## 1692          Hütt Kristo 13668          30
## 1777         Keerdo Kaivo 13856          31
## 1778         Kuslap Handri 13856          31
## 2050         Vlassov Jüri 14441          32
```

```
## 2051        Hion Lars-Erik 14442         32
## 2379           Tsirp Priit 15346         33
## 2380     Allilender Rando 15346         33
## 2387           Hints Kairi 15361         34
## 2388         Haldre Henri 15363         34
## 2632       Nõmmiste Kalev 16170         35
## 2633       Nõmmiste Sulev 16170         35
## 2644      Tiedemann Tõnis 16237         36
## 2645         Talvik Heiki 16237         36
## 2657           Raud Ander 16320         37
## 2658         Kurvits Erko 16320         37
## 2681         Märss Martin 16388         38
## 2682     Lõhmus Ann-Marii 16388         38
## 2702           Karbe Sven 16517         39
## 2705       Schults Markko 16524         39
## 2773          Ennok Brita 16918         40
## 2774         Tarjus Piret 16918         40
## 2797        Visnap Thomas 17066         41
## 2798      Pähklamäe Ville 17066         41
## 2802         Rebane Urmas 17075         42
## 2803          Valge Kerli 17075         42
## 2842   Dombrovskis Mareks 17318         43
## 2843    Dombrovska Jelena 17319         43
## 2876       Tenisson Vaido 17695         44
## 2877      Tenisson Silvia 17695         44
## 2947         Raasik Kaire 18617         45
## 2948         Raasik Marko 18617         45
## 2955        Võikar Raitel 18740         46
## 2956             Unt Siim 18740         46
## 2971       Punane Krista 19029          47
## 2972         Punane Urmas 19029         47
## 3025           Kaasik Lea 20979         48
## 3026        Kaasik Margus 20979         48
## 3038          Rähni Ringo 21986         49
## 3039         Rähni Markus 21986         49
## 3044          Paide Tanel 22177         50
## 3045   Paide Jarl Patrick 22177         50
## 3047         Veeroja Liis 22422         51
## 3048        Külaots Urmet 22422         51
## 3056          Roots Urmas 23179         52
## 3057  Pruulmann Annemaria 23180         52
```

Clustering relevance - show that people who were part of a pace group were more consistent, experienced and had better results.

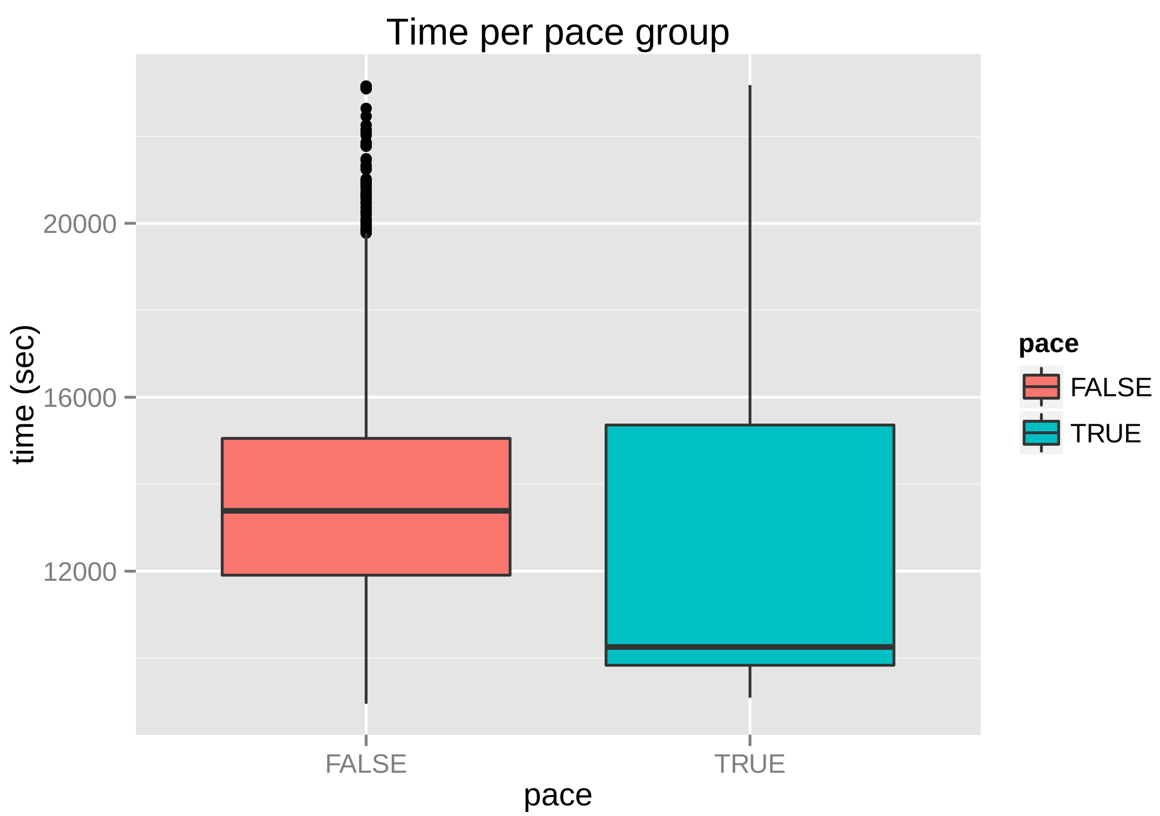
```r
#Average cluster size - i.e how big the pace groups were on average?
round(mean(table(dsData[,"ds$cluster"])[-1]),0)
```

```
## [1] 3
```

```r
solo = dsData[dsData[,"ds$cluster"] == 0, "time"]
pace = dsData[dsData[,"ds$cluster"] > 0, "time"]
t.test(solo, pace)
```

```
##
##  Welch Two Sample t-test
##
## data:  solo and pace
## t = 3.81, df = 152.32, p-value = 0.0002012
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   596.1903 1880.4403
## sample estimates:
## mean of x mean of y
##  13679.24  12440.93
```

```
dsData$pace = dsData[,"ds$cluster"] > 0
ggplot(dsData, aes(x = pace, y = time, fill = pace)) +
  geom_boxplot() +
  labs(title = "Time per pace group", y = "time (sec)")
```



Result: people who were part of a pace group had better results than people who went solo.

```
# % of people who ride in pace group
length(pace)*100/(length(solo)+length(pace))
```

```
## [1] 4.841348
```

```
# % of the people in pace group who had completed
# at least 1 marathon before
nrow(dsData[dsData[,"ds$cluster"]>1 & dsData[,"particip.time"] > 1, ])*100/length(pace)
```

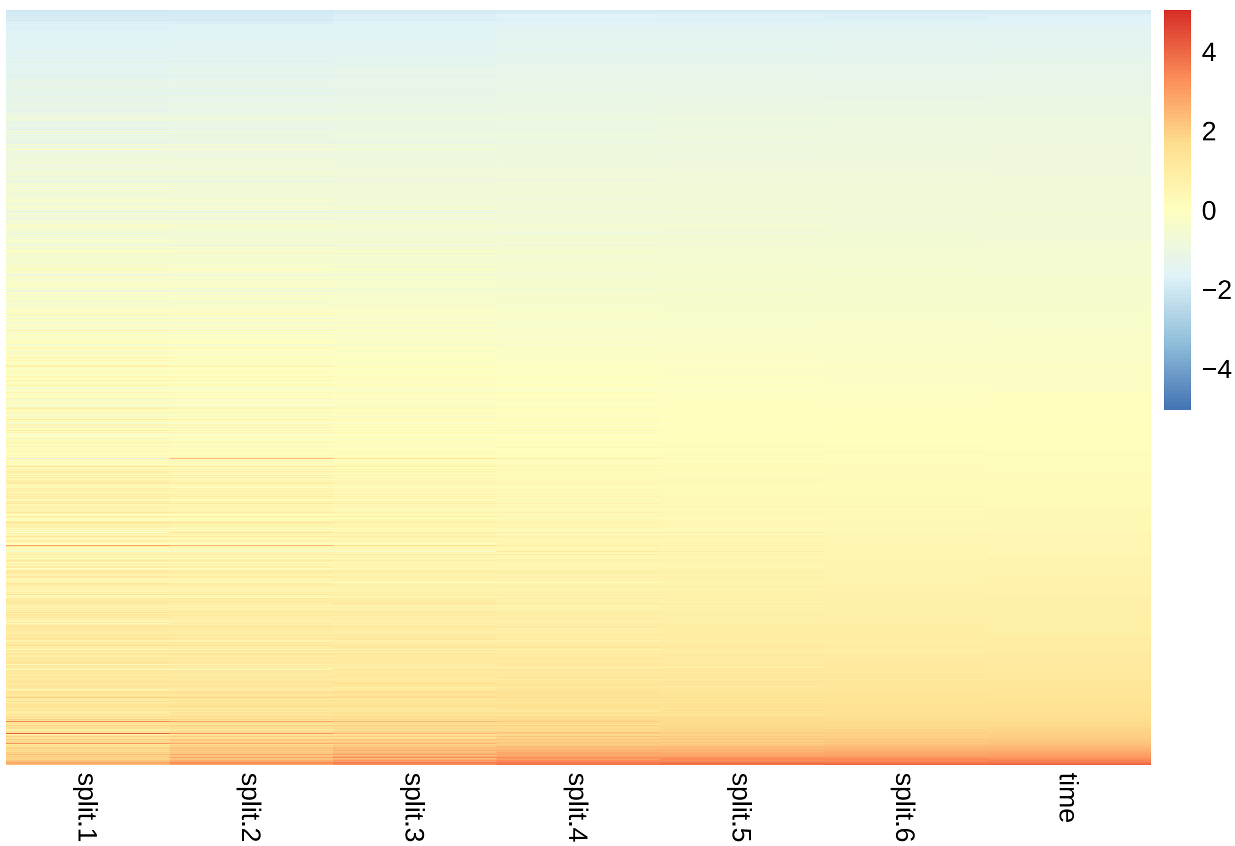
```
## [1] 87.83784
```

Clustering visualization

Overall heatmap

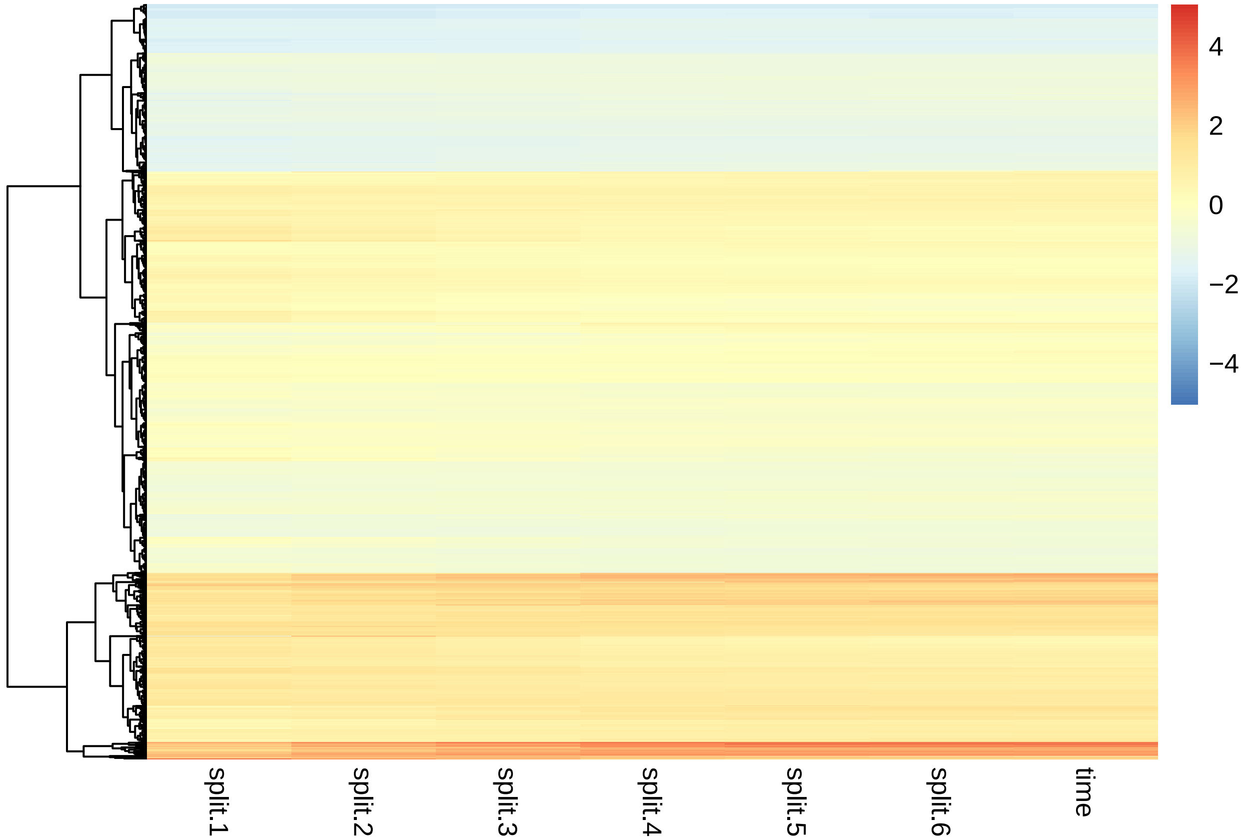
Normalize by columns to make the splits. Comparable, since time increases with each split.

```
pheatmap(data[,c(paste("split.",1:6,sep=""), "time")], cluster_rows = F,
         cluster_cols = F, scale="column", show_rownames = F)
```



Clustered heatmap - hierarchical

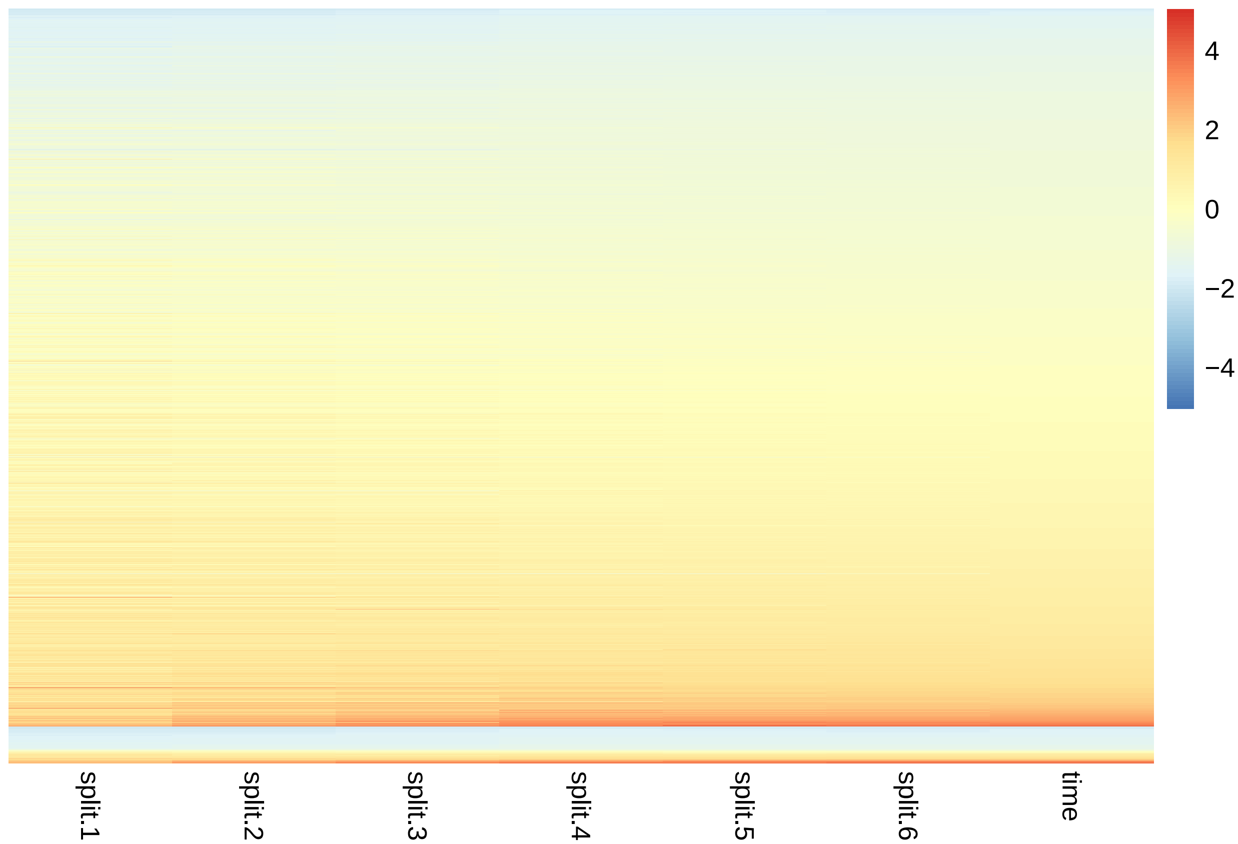
```
pheatmap(data[,c(paste("split.",1:6,sep=""), "time")], cluster_rows = T,
         cluster_cols = F, scale="column", show_rownames = F)
```



Clustered heatmap - dbscan

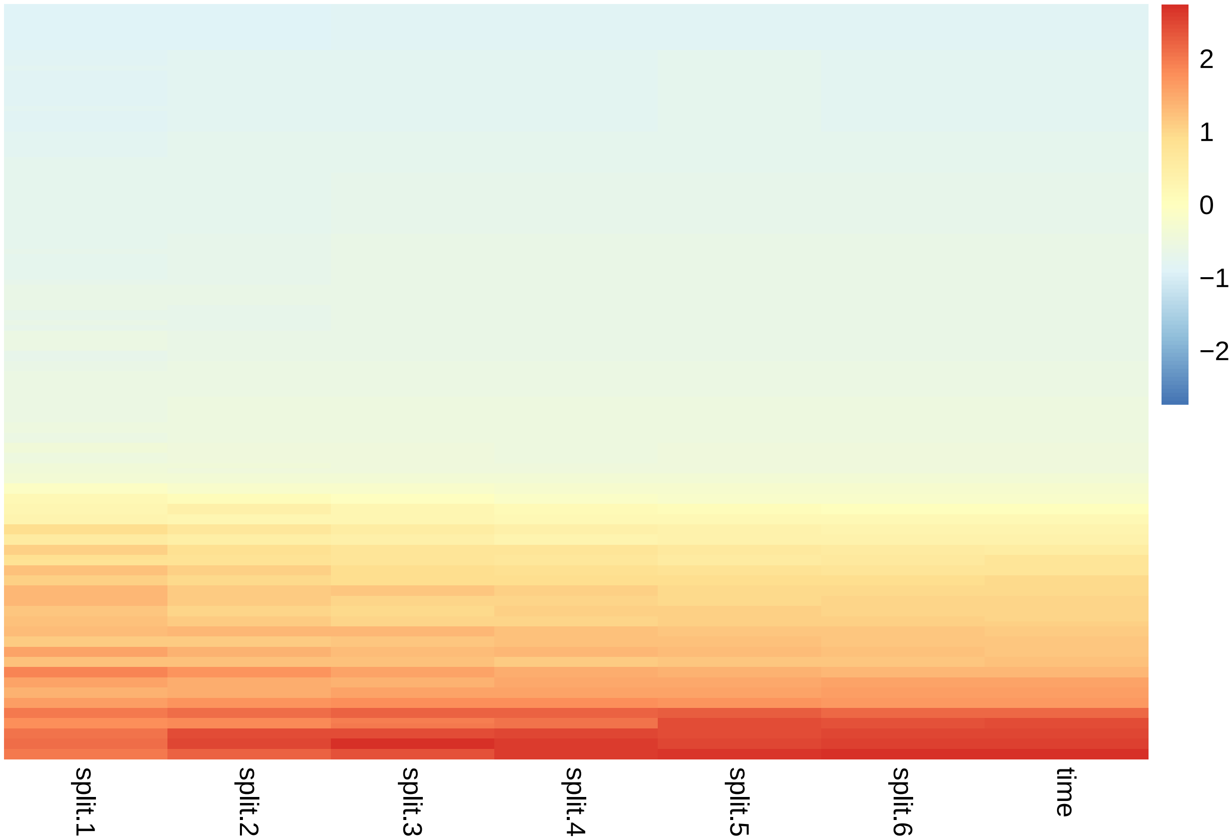
Clusters appear at the bottom, rest is noise.

```
pheatmap(dsData[,c(paste("split.",1:6,sep=""), "time")], cluster_rows = F,
         cluster_cols = F, scale="column", show_rownames = F)
```



Heatmap of dbscan clusters

```
pheatmap(dsData[dsData[,"ds$cluster"] > 0 ,c(paste("split.",1:6,sep=""), "time")],
         cluster_rows = F, cluster_cols = F, scale="column", show_rownames = F)
```



Regression

Regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

```
#Read in data
data = read.table("data/processedData.txt", header=T)
```

Model

In this case the dependent variable is finishing time and the independent variables are age, country, starting number and participation time which are used for generating a multiple linear regression model.

```
#Fit model using using linear model
lmfit <- lm(timeCategory ~ ageCategory + countryCategory + sNrCategory + participTimeCategory, data=data
form <- as.matrix(coef(lmfit))
rownames(form) <- gsub("try", "try == ", rownames(form) )
rownames(form) <- gsub("oup", "oup == ", rownames(form) )
rownames(form)[1] <- "Base"
cat(paste( form, paste("(", rownames(form), ")" ), sep="*", collapse="+\n") )
```

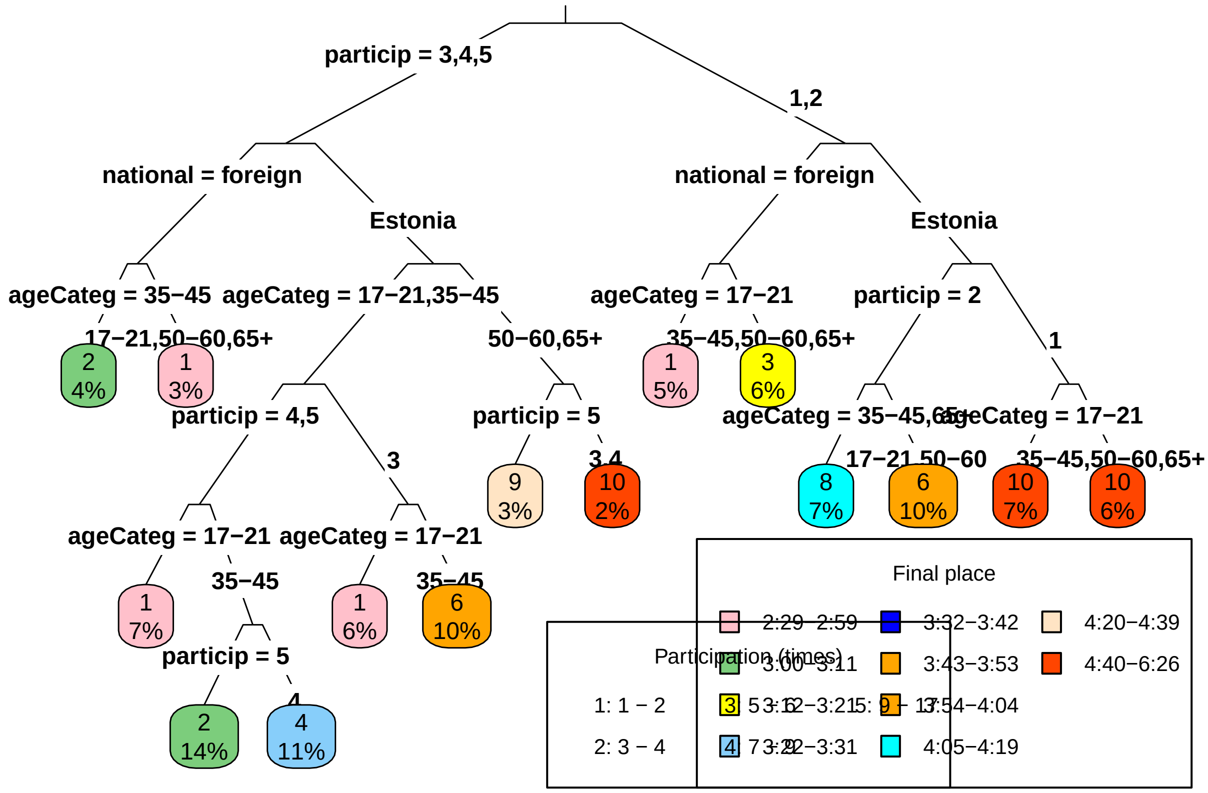
```
## 0.0653330397190914*( Base )+
## 0.167323137488844*( ageCategory35-45 )+
```

```
## 0.630270746880228*( ageCategory50-60 )+
## 1.6405431331373*( ageCategory65+ )+
## 3.49025506483708*( country == CategoryHiina )+
## -0.0433780713320298*( country == CategoryIiri )+
## 0.947807275050404*( country == CategoryInglismaa )+
## -1.51478143557452*( country == CategoryLeedu )+
## -1.16352004588815*( country == CategoryLäti )+
## 0.582954812623443*( country == CategoryNorra )+
## -2.0551201024706*( country == CategoryRootsi )+
## -4.05219272494975*( country == CategorySaksamaa )+
## -0.249325802274713*( country == CategorySoome )+
## -1.70626377469707*( country == CategoryTaani )+
## -0.83189518437217*( country == CategoryVenemaa )+
## 1.54244778978653*( sNrCategory )+
## 0.274620736297802*( participTimeCategory )
```

Decision tree

Each branch represents the outcome of the test and each leaf node represents a class (decision taken after computing all attributes).

```
#Decision tree
data$participTimeCategory = as.factor(data$participTimeCategory)
fit <- rpart(timeCategory ~ ageCategory + nationality + participTimeCategory, method="class", minbucket
colors <- c("pink", "palegreen3", "yellow", "LightSkyBlue", "blue", "orange", "orange", "cyan", "bisque
boxcols <- (colors)[fit$frame$yval]
prp(fit, type=3, extra=100, faclen = 0, cex = 0.75, box.col = boxcols)
legend("bottomright", xpd = TRUE, inset = c(0, 0), cex = 0.7, ncol=3, fill = colors, title="Final place
       legend = c("2:29-2:59",
                  "3:00-3:11",
                  "3:12-3:21",
                  "3:22-3:31",
                  "3:32-3:42",
                  "3:43-3:53",
                  "3:54-4:04",
                  "4:05-4:19",
                  "4:20-4:39",
                  "4:40-6:26"
                  ))
legend("bottomleft", xpd = TRUE, inset = c(0.45, 0), cex = 0.7, ncol=3, title="Participation (times)",
       legend = c("1: 1 - 2",
                  "2: 3 - 4",
                  "3: 5 - 6",
                  "4: 7 - 9",
                  "5: 9 - 17"
       ))
```



Reference

- Wikipedia