ReproRehab Week 3: Clean and Merge

Andrew Hooyman

2024-10-20

1.1 Clean and Merge

In week 2 we combined the individual raw data files into one large "parent" file. We can now begin to merge this parent file with other data, such as demograpics, so we can progress to eventual analysis.

Right now the parent file is in long format, i.e. many varibles nested within it, trial, sub, etc.

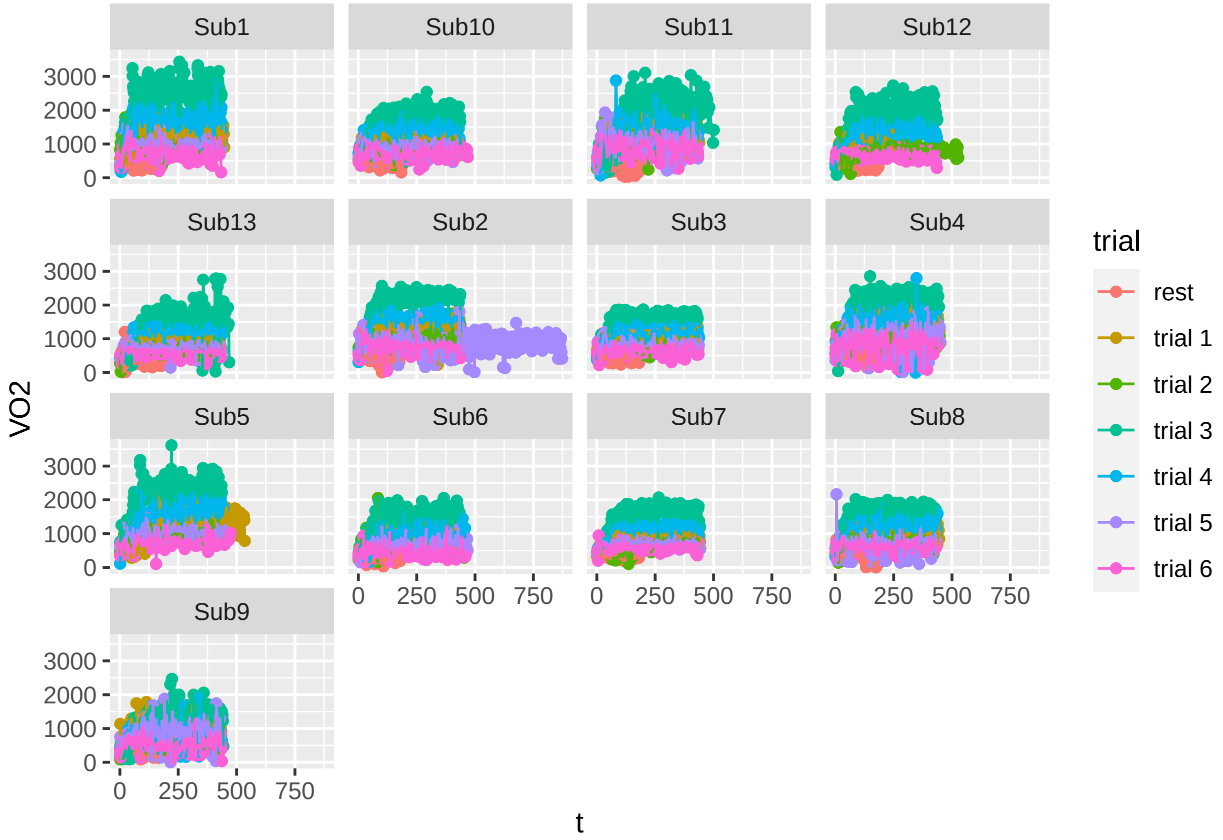
For this lesson we are going to look at: 1. Do some basic cleaning. 2. methods to convert it back to wide, and then back to long again. 3. Merge with a separate demographic data set. 4. Produce an aggregate of the parent data set. 5. Use plotting as our "sanity check"

1.1 Basic Cleaning

```
library(here)
```

```
## here() starts at C:/Users/ahooyman/ASU Dropbox/Andrew Hooyman/ReproRehab/2024/Clean and Merge week 3
```

```
library(ggplot2)

#I previously saved the parent data set as a .csv
#Reading it in here
data.all=read.csv("raw.data.all.csv")

#Visualize again, across the time series by subject
ggplot(data.all,aes(x=t,y=VO2,color=trial))+
  geom_point()+
  geom_line()+
  facet_wrap(~Sub)
```



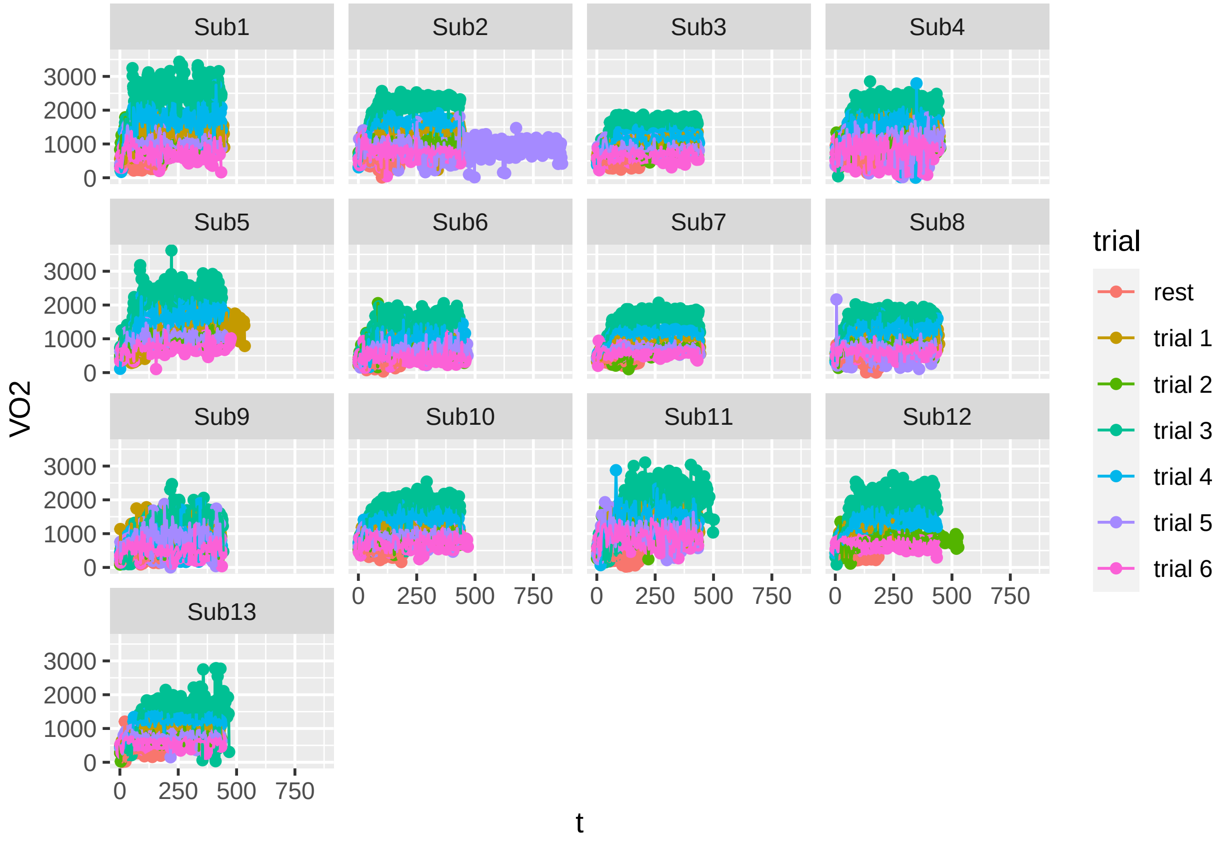
```
#The first problem is that R still thinks my subjects are listed based on the
#value of the first integer. We need to change this first so it displays
#numerically.

#Right now the Sub variable is a character variable class
class(data.all$Sub)
```

```
## [1] "character"
```

```
#But I can convert it to a factor and then set the levels of the factor into
#the order I want them to be.
data.all$Sub=factor(data.all$Sub,levels = c("Sub1","Sub2","Sub3","Sub4",
                                            "Sub5", "Sub6","Sub7","Sub8",
                                            "Sub9","Sub10","Sub11","Sub12",
                                            "Sub13"))
#This is more desireable anyway because any variable of class character is
#implicitly converted to a factor when run through an lm or lme anyway.
class(data.all$Sub)
```

```
## [1] "factor"
```

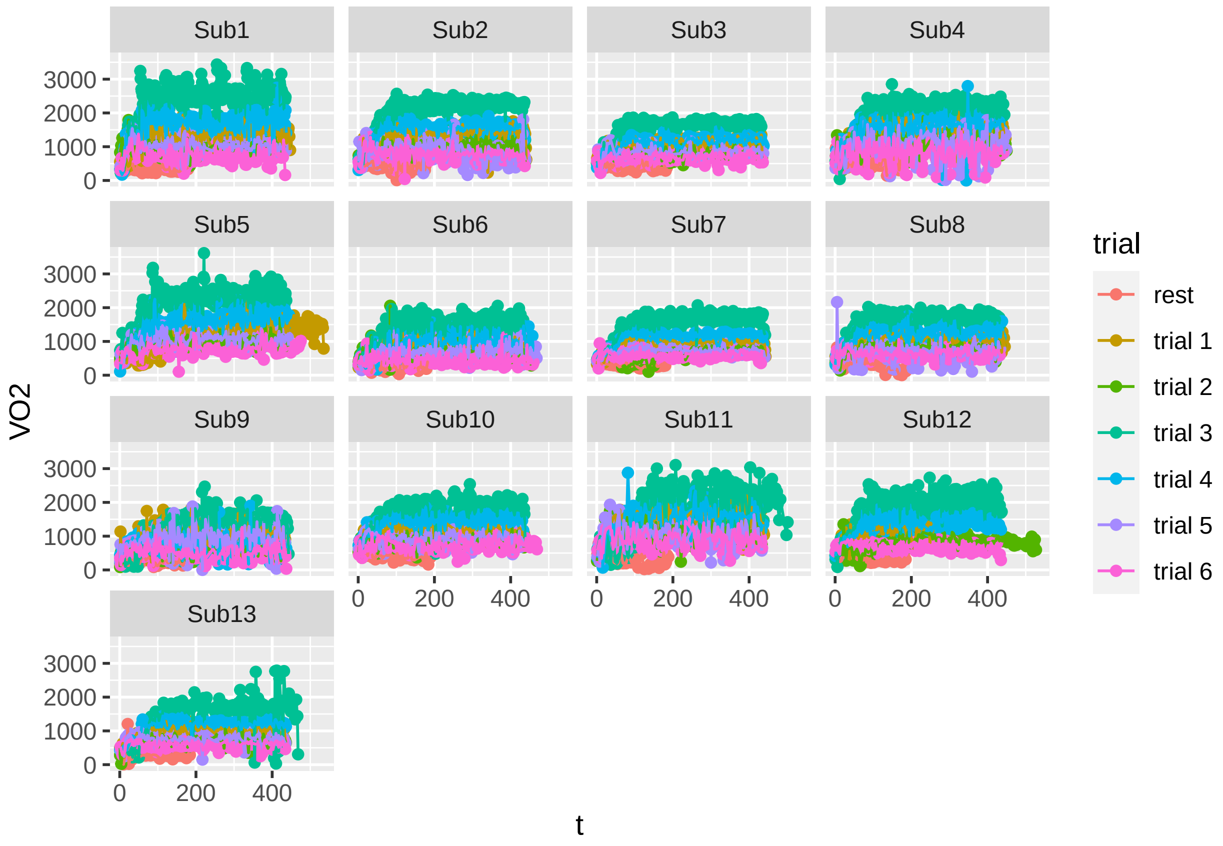
```
#Fixed
ggplot(data.all,aes(x=t,y=VO2,color=trial))+
  geom_point()+
  geom_line()+
  facet_wrap(~Sub)
```



```r
# Sub 2 did way more time than everyone else on trial 5. I am going to assume
#that doesn't warrant throwing that subjects data out, and instead I will just
#trim the excess away.

#Let's first see how much of the extra to remove. Let's look at how long
#all subject's trial 5 was.

#The function aggregate will provide some summary stat based on how I want
#the dataset aggregated. Here I just want the max time for trial 5 across all
#subjects. To reduce the output I just index the data.all to only include trial 5.
#Looks like Sub 2 did 873 seconds, and everyone else did ~440
aggregate(t~Sub,data.all[data.all$trial=="trial 5",],max)
```

```
##      Sub   t
## 1   Sub1 436
## 2   Sub2 873
## 3   Sub3 437
## 4   Sub4 449
## 5   Sub5 443
## 6   Sub6 468
## 7   Sub7 441
## 8   Sub8 433
## 9   Sub9 433
## 10 Sub10 436
## 11 Sub11 433
## 12 Sub13 435
```

```
#I am going to remove all t for Sub 2 greater than 440
#I am going to use a method called negation !()
#Here I put in a bunch of boolean arguments into () and then preceed it with a !
#This removes all the rows in my dataset that match these conditions.
data.all2=data.all[!(data.all$trial=="trial 5" & data.all$Sub=="Sub2" & data.all$t>440),]

#If I removed the !() it would only return the trial 5, for Sub2 with t > 440

#Fixed
ggplot(data.all2,aes(x=t,y=VO2,color=trial))+
  geom_point()+
  geom_line()+
  facet_wrap(~Sub)
```



```
#I could repeat this process again but let's move on.
```

1.2 Long to wide

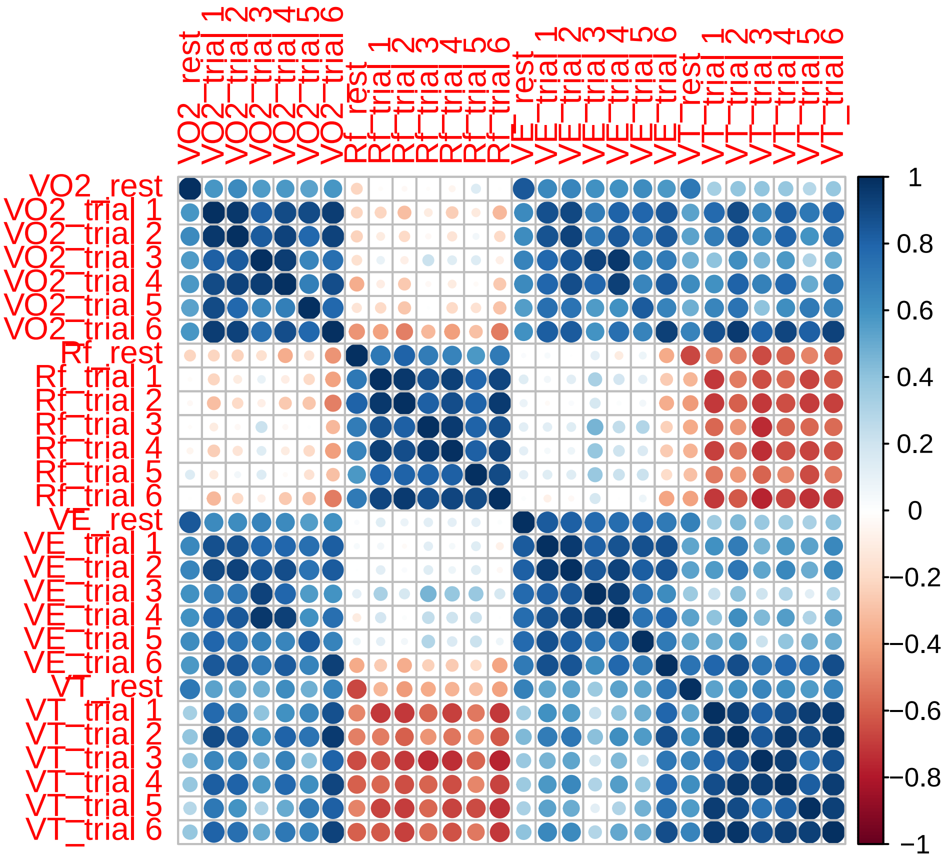
Now I am going to convert the data from long to wide format. This can be useful if I want to look at correlations between my behavioral variables.

```
library(tidyr)

#first I am going to create an aggregate of the data
```

```
data.agg=aggregate(cbind(VO2,Rf,VE,VT)~Sub+trial,data.all2,mean)

#This created a summarized data set across our 4 outcome variables average across all time for each sub

#I am going to use the tidyr library to "pivot" my data frame along the trial variable.

#Now I am going to use the pivot_wider function from tidyr to create a new variable for each trial
data.agg.w=pivot_wider(data.agg,names_from = trial,
                       values_from = VO2:VT,id_cols = Sub)

#create a correlation matrix with only comlete observations,
#Can only include numeric variables.
cmat=cor(data.agg.w[,-1],use="complete.obs")

#Library for correlation plot
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#Let's visualize the correlations.
corrplot(cmat)
```

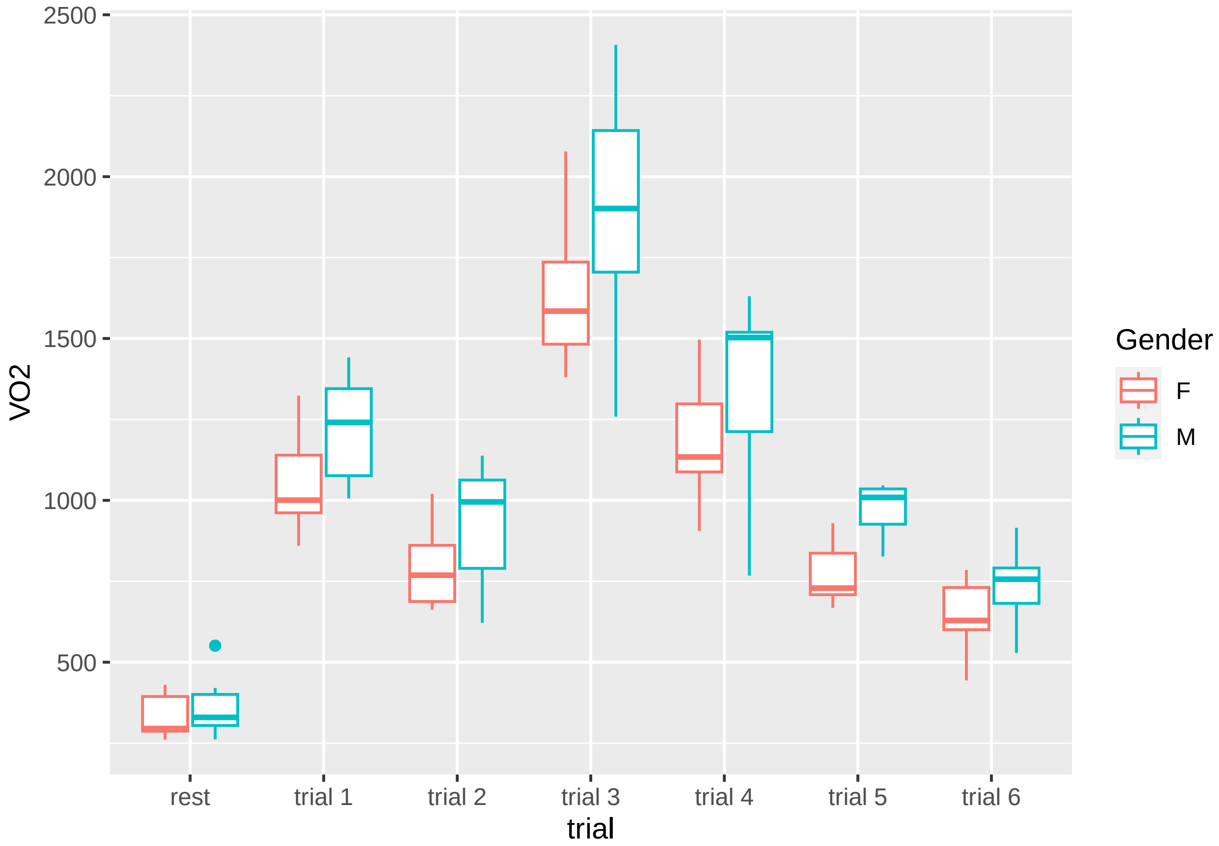


```
#Now let's convert our wide dataset back to long, just for fun.
data.agg.l <- pivot_longer(data.agg.w,
                           cols = VO2_rest:`VT_trial 6`,
                           names_to = c(".value", "trial"),
                           names_sep = "_")
```

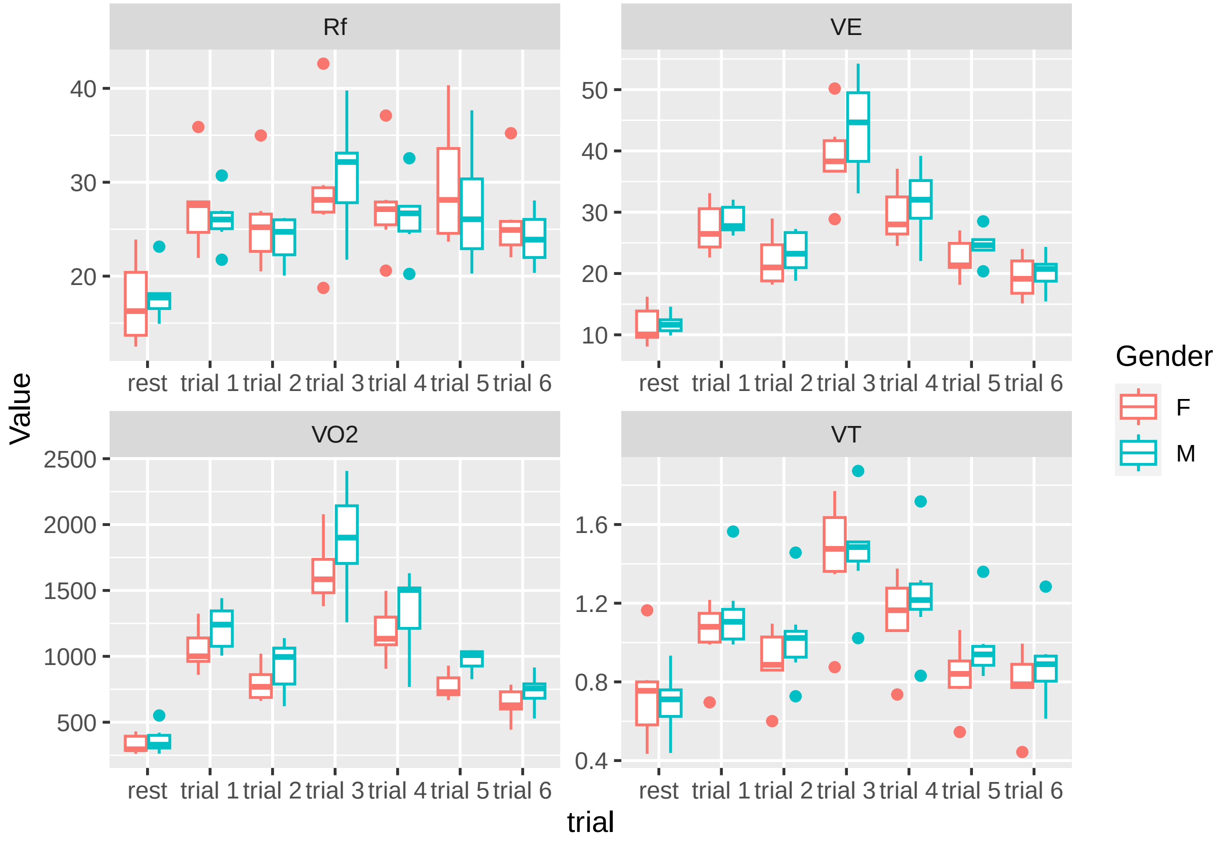
1.3 Merge with Demographics

Now that we have our data in a variety of formats we can now merge it with the demographic data.

```
library(readxl)

demo=read_excel("SubjectInfo.xlsx")

#merging the demo data to the aggregate data in the long format
#If the names of the merge variable are different then we have to call that explicitly
demo.data=merge(data.agg,demo,by.x="Sub",by.y = "Subject No")

#Have to order the levels of the trial
demo.data$trial=factor(demo.data$trial,levels = c("rest","trial 1","trial 2",
                                                  "trial 3","trial 4","trial 5","trial 6"))

ggplot(demo.data,aes(x=trial,VO2,color=Gender))+
  geom_boxplot()
```



```
#Let's pivot the aggregate data again to look at all variables in one column

demo.data.var.long=pivot_longer(demo.data,
                                cols = VO2:VT,
                                names_to = "Variable",
                                values_to = "Value")

ggplot(demo.data.var.long,aes(x=trial,Value,color=Gender))+
  geom_boxplot()+
  facet_wrap(~Variable,scales = "free")
```


Rf
VE
VO2
VT
rest
trial 1
trial 2
trial 3
trial 4
trial 5
trial 6
Value
trial
Gender
F
M