

# Longitudinal Data Analysis

Case study of Trenal.XLS using Linear Mixed Effect Model

Group2: Wanchang Zhang; Hugo Blain; Oscar Cabanelas

2023-03-22

## Contents

<b>Theory of Linear Mixed Effects Model(LMM)</b>	<b>3</b>
Index description . . . . .	3
The application of LMM . . . . .	3
<b>Data set Trenal.XLS pre-analysis</b>	<b>3</b>
The summary of the data set . . . . .	3
Import data . . . . .	3
Data Preprocessing . . . . .	3
Response variable and predictors . . . . .	4
Data visualization and the information from the data . . . . .	5
<b>Data set Trenal.XLS analysis with the linear mixed effects model</b>	<b>18</b>
The chosen of fixed effects variable . . . . .	18
Including Plots . . . . .	19

## List of Figures

## List of Tables

1. Describe the data, and use graphical techniques to explore the mean structure, the variance structure and the correlation structure. Summarize your conclusions. What are the implications with respect to statistical modeling?
2. What summary statistics are appropriate for the analysis of these data? Why? Do they yield the same results? Summarize your conclusions.
3. Fit a multivariate model and find the most parsimonious mean structure which can be used to describe the average evolutions in the data. What covariance structures are applicable in this case? What is the most parsimonious structure you can find?
4. Use an explicit two-stage analysis to get an initial impression about trends and effects of covariates.
5. Formulate a plausible random-effects model. Fit your model and compare the results with those from the multivariate model. Check the appropriateness of your random-effects model. Calculate the subject-specific intercepts/slopes and compare them with the ones you obtained from a two-stages analysis. What do you conclude?
6. Pay attention to the missing, especially the ones presented by the outcome variable. Do your results still hold despite the missingness?

## Theory of Linear Mixed Effects Model(LMM)

### Index description

Let us assume that a given input data set  $X$  has a dimension  $N \times p$ , with  $N$  observations and  $p$  predictors. For each subject indexed with  $i, i = 1, \dots, I$ , we can build a linear mixed effect model

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \epsilon_i$$

### The application of LMM

Linear Mixed Effects Model is used to analyse a data set, where the observations are not fully independent, while the top level clusters are assumed independent. Inside each cluster, the observations are correlated

## Data set `Trenal.XLS` pre-analysis

### The summary of the data set

#### Import data

```
library(readxl)
trenal <- read_excel("Trenal.XLS") # summary(trenal)
```

#### Data Preprocessing

```
trenal= trenal[,-18] #remove a noninformative column const
# Continuous or discrete variables
trenal$id = as.factor(trenal$id)
trenal$j = as.factor(trenal$j)
trenal$male = as.factor(trenal$male)
trenal$cardio = as.factor(trenal$cardio)
trenal$reject = as.factor(trenal$reject)

trenal.long = trenal[,13:20] # long table form
trenal.wide = trenal[,1:17] # wide table form
```

```

library(magrittr) # needs to be run every time you start R and want to use %>%
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

data.long <- trenal.long %>% # reordered long table
  relocate(id) %>%
  relocate(j,.after=id)%>%
  relocate(time,.after = j)%>%
  relocate(respons,.after=time)
#summary(data.long)
sum(!is.na(data.long$respons))

## [1] 9558

data.long.noNA <- na.omit(data.long)# reordered long table without NAs
summary(data.long.noNA)

##           id           j           time           respons           age
##  3      : 12    2      :1159   Min.    : 0.000   Min.    :14.00   Min.    :15.00
##  5      : 12    1      :1158   1st Qu.: 1.000   1st Qu.:34.00   1st Qu.:35.00
##  6      : 12    3      :1158   Median : 3.000   Median :38.00   Median :46.00
##  8      : 12    4      :1072   Mean    : 3.432   Mean    :38.24   Mean    :45.27
##  9      : 12    5      : 954   3rd Qu.: 6.000   3rd Qu.:42.00   3rd Qu.:56.00
## 10      : 12    6      : 845   Max.    :10.000   Max.    :65.00   Max.    :76.00
## (Other):9479   (Other):3205
## male      cardio  reject
## 0:4213    0:7927   0:6314
## 1:5338    1:1624   1:3237
##
##
##
##
##

data.long.noNA$id[length(data.long.noNA$id)]

## [1] 1160
## 1160 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 1160

```

## Response variable and predictors

**Response variable** From the `summary(data.long.noNA)`, we can read that the response variable is a continuous variable `respons` HC level from (15,76) with Mean 38.24.

We have totally  $I = 1160$  ids for subjects. Ideally each id would have 12 (start from  $HC_0, HC_{0.5}, HC_1, HC_2, \dots, HC_{10}$ ) HC level measurement, but in reality not all subjects have all of the 12 measurements. We have totally  $N = 9558 = \sum_{i=1}^I n_i$  missing values.

**Predictors** The explaining variables are

1.  $X_1 = \text{time}$  in year as discrete values, only changes with  $j, j = 1, \dots, n_i$
2.  $X_2 = \text{age}$  in year with 12 NAs; will only change with subject id  $i$
3.  $X_3 = \text{male}$  0 = female, 1 = male; will only change with subject id  $i$
4.  $X_4 = \text{cardio}$  0 = no, 1 = yes; will only change with subject id  $i$
5.  $X_5 = \text{reject}$  0 = accept, 1 = reject; will only change with subject id  $i$
6. fixed intercept, continuous

In the data analysis part, we need to try out different variables accounting for fixed effects and random effects.

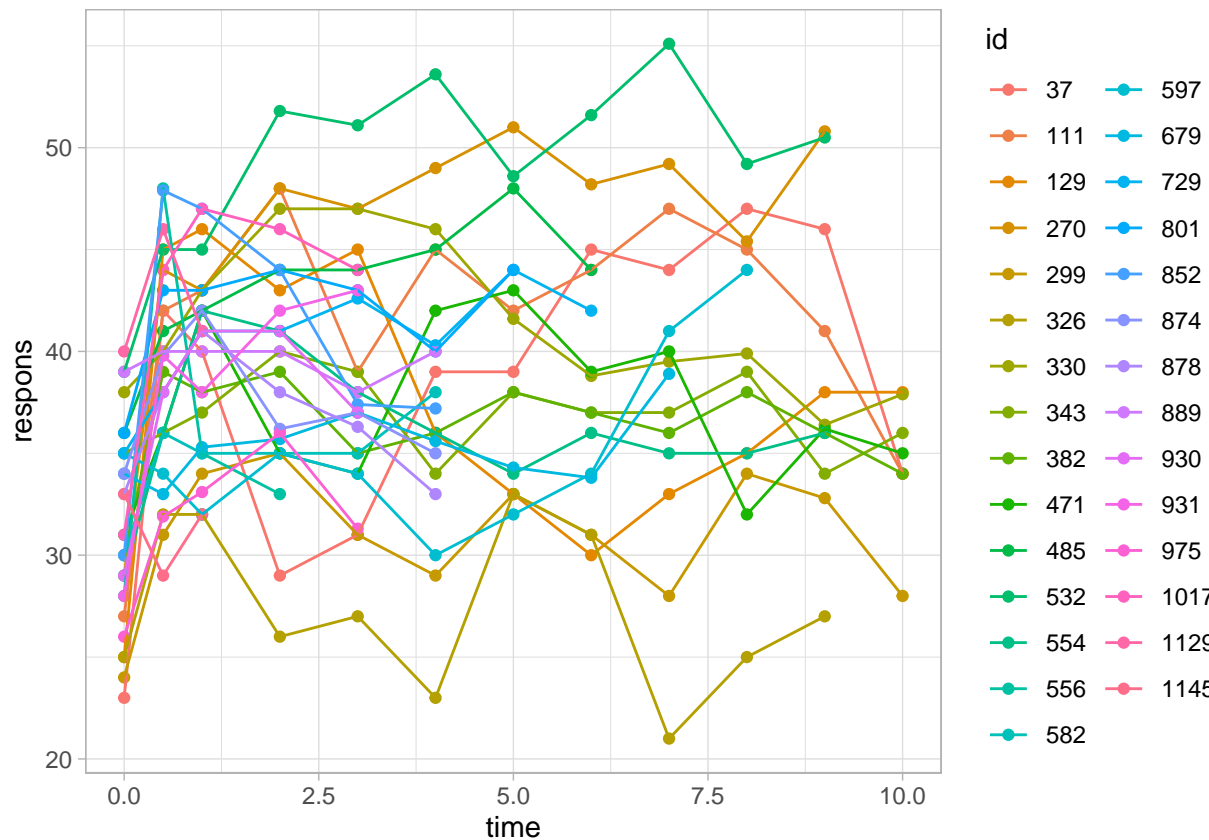
### Data visualization and the information from the data

```
library(ggplot2)
dim(data.long.noNA)
```

```
## [1] 9551      8
```

```
# since the data dimension is large 9551 x 8, we can select random 30 data to have a look
set.seed(1)
selected <- sample(1:length(unique(data.long.noNA$id)),30,replace=T) # random samples and permutations
#selected.vector = as.vector(selected)
data.selected = data.long.noNA[(data.long.noNA$id %in% c(selected)), ]
```

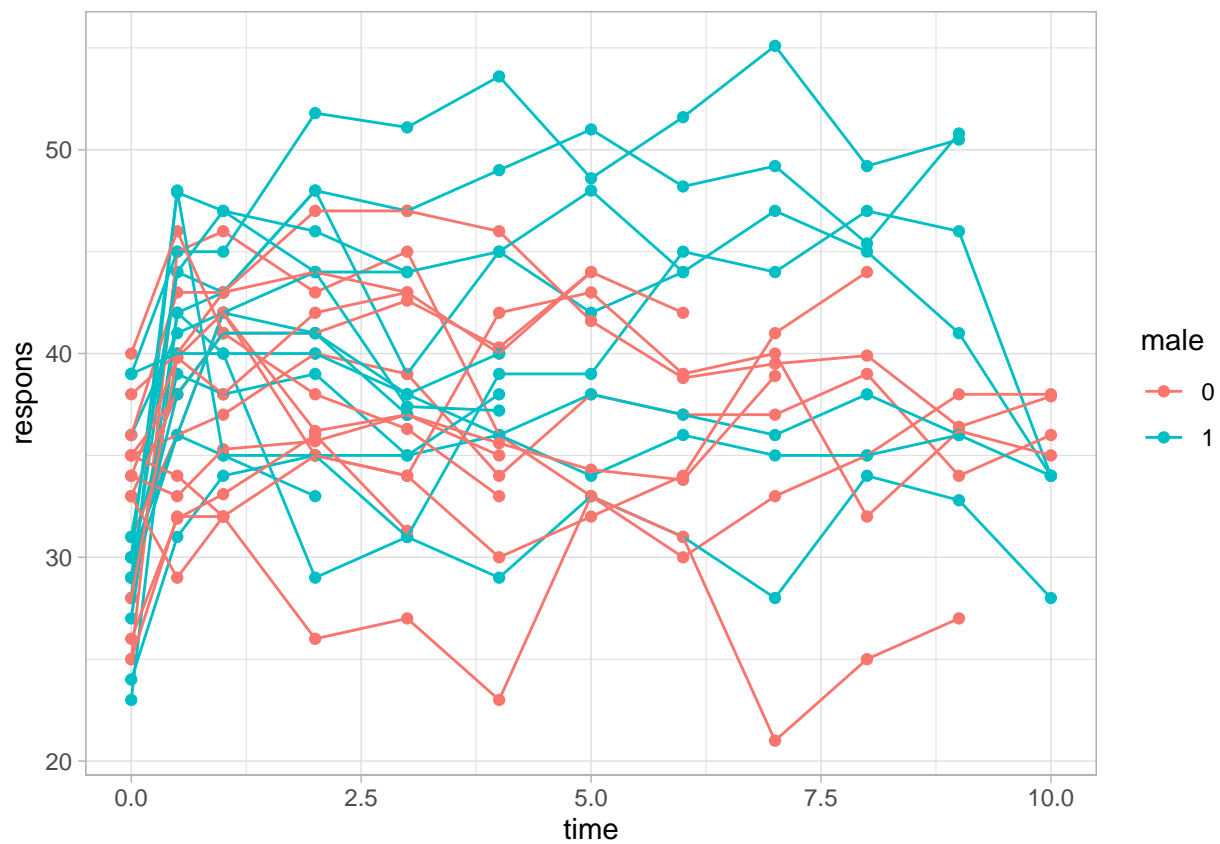
```
ggplot(data.selected,aes(x=time,y=respons,group=id,color=id))+geom_point()+ geom_line()+theme_light()
```



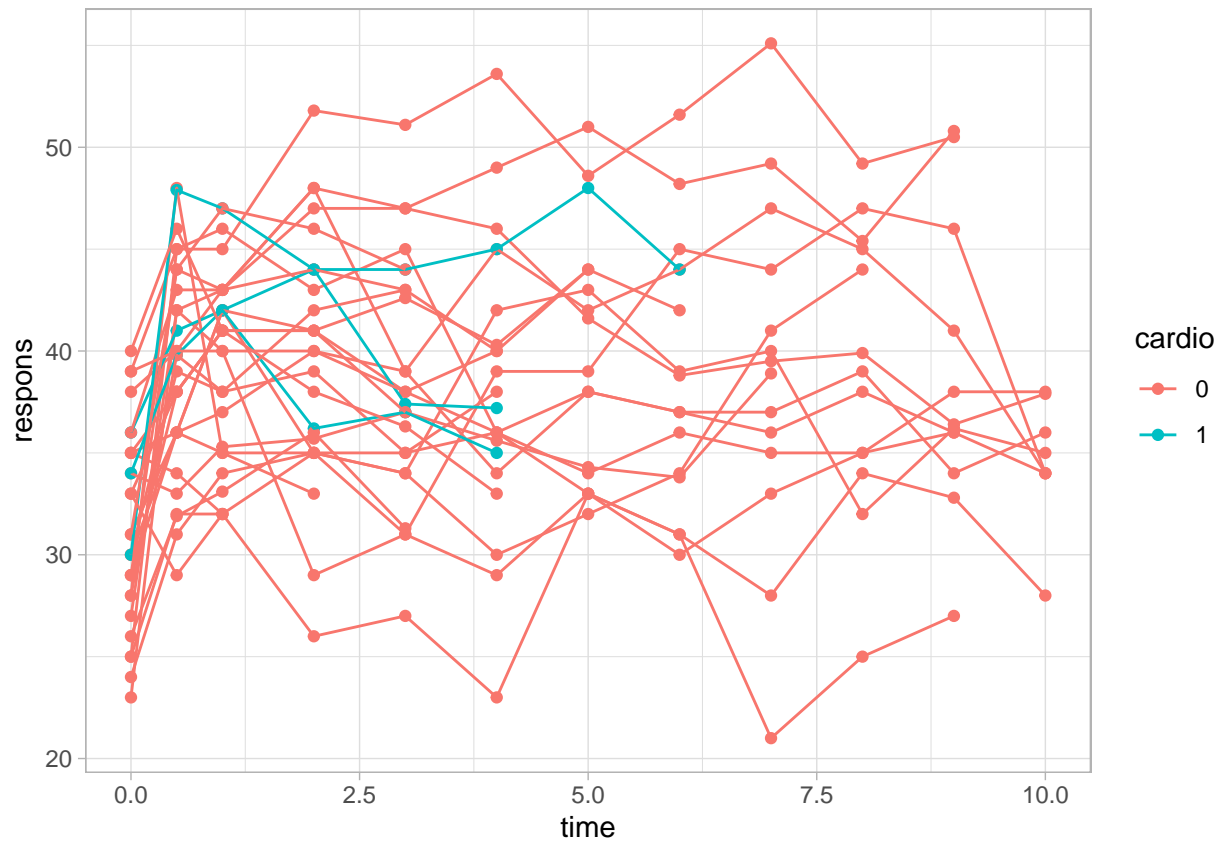
### Spaghettic plot

Plot Info: The intercept may vary according to each individual The slope is not very easy to see

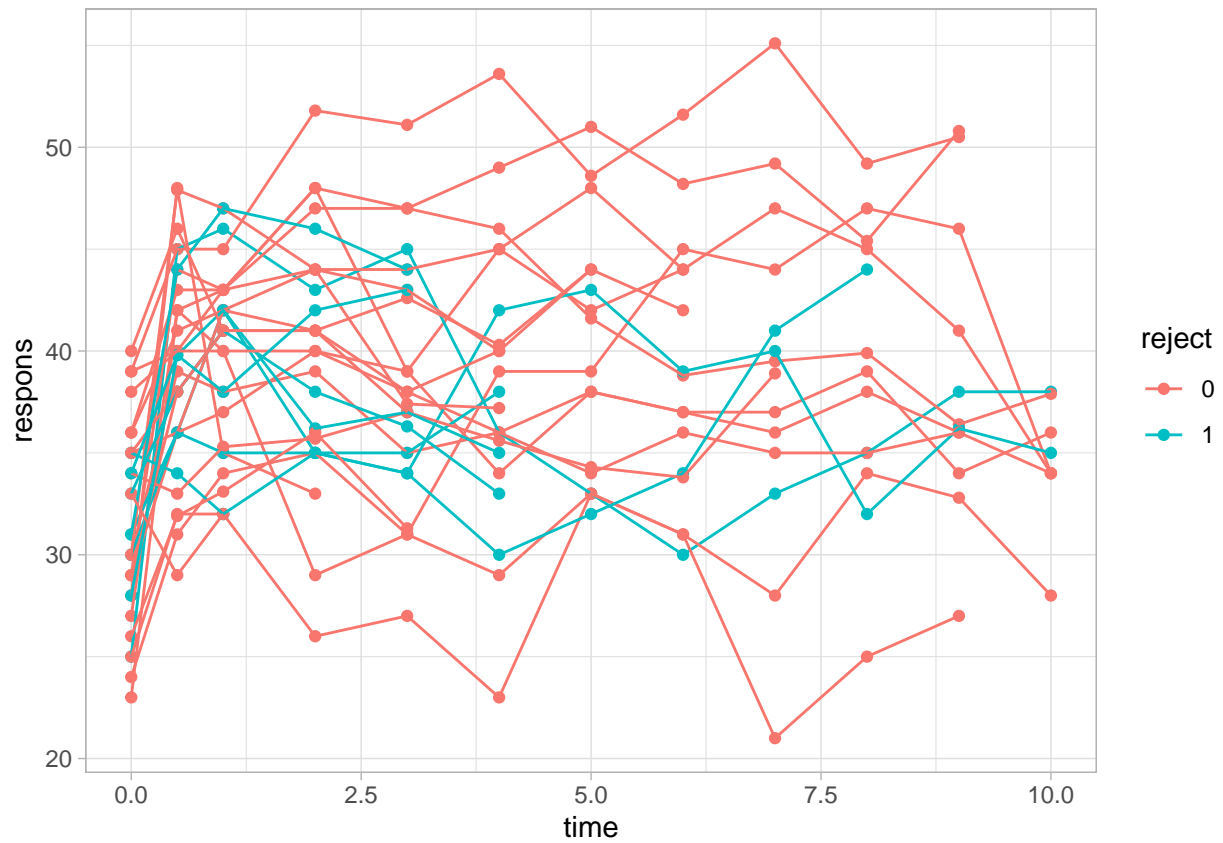
```
ggplot(data.selected,aes(x=time,y=respons,group=id,color=male)) +geom_point()+ geom_line()+theme_light()
```



```
ggplot(data.selected,aes(x=time,y=respons,group=id,color=cardio)) +geom_point()+ geom_line()+theme_light()
```

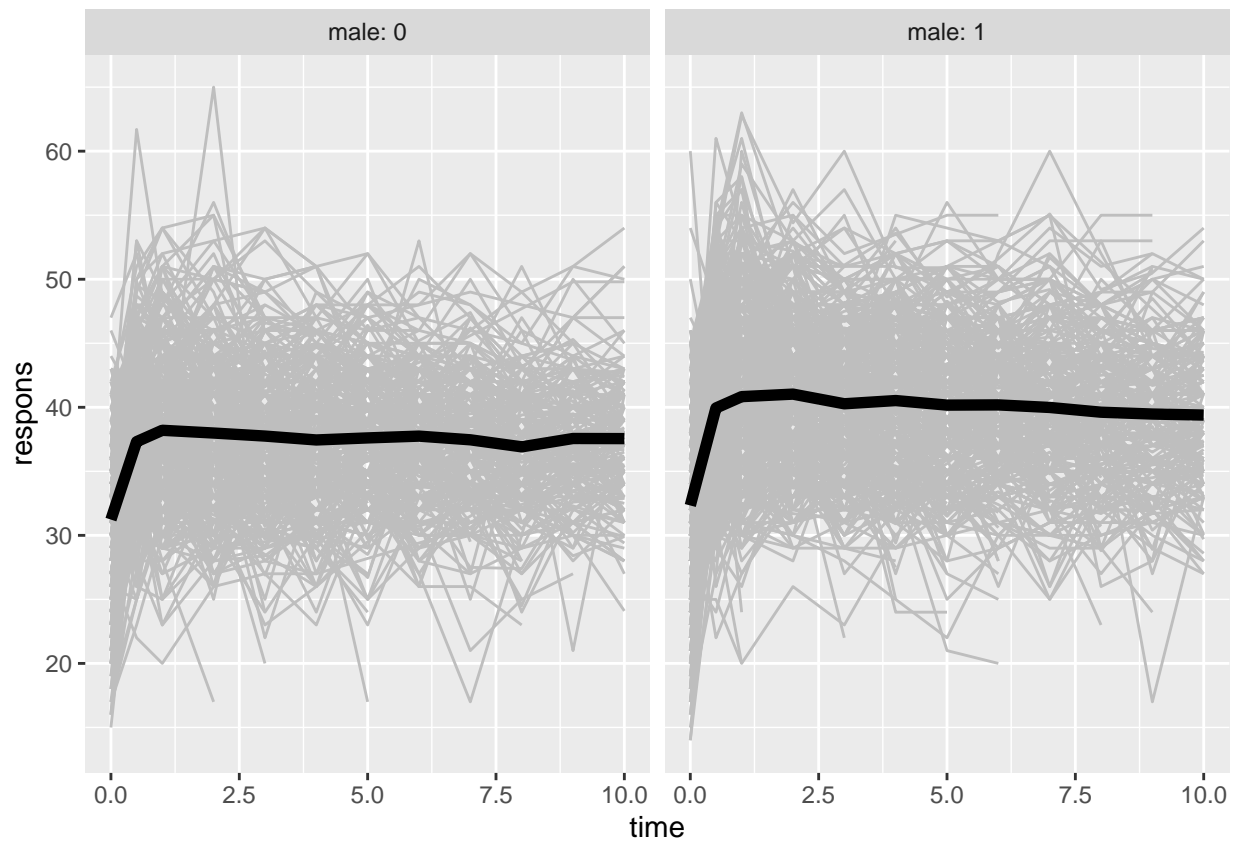


```
ggplot(data.selected,aes(x=time,y=respons,group=id,color=reject)) +geom_point()+ geom_line()+theme_ligh
```

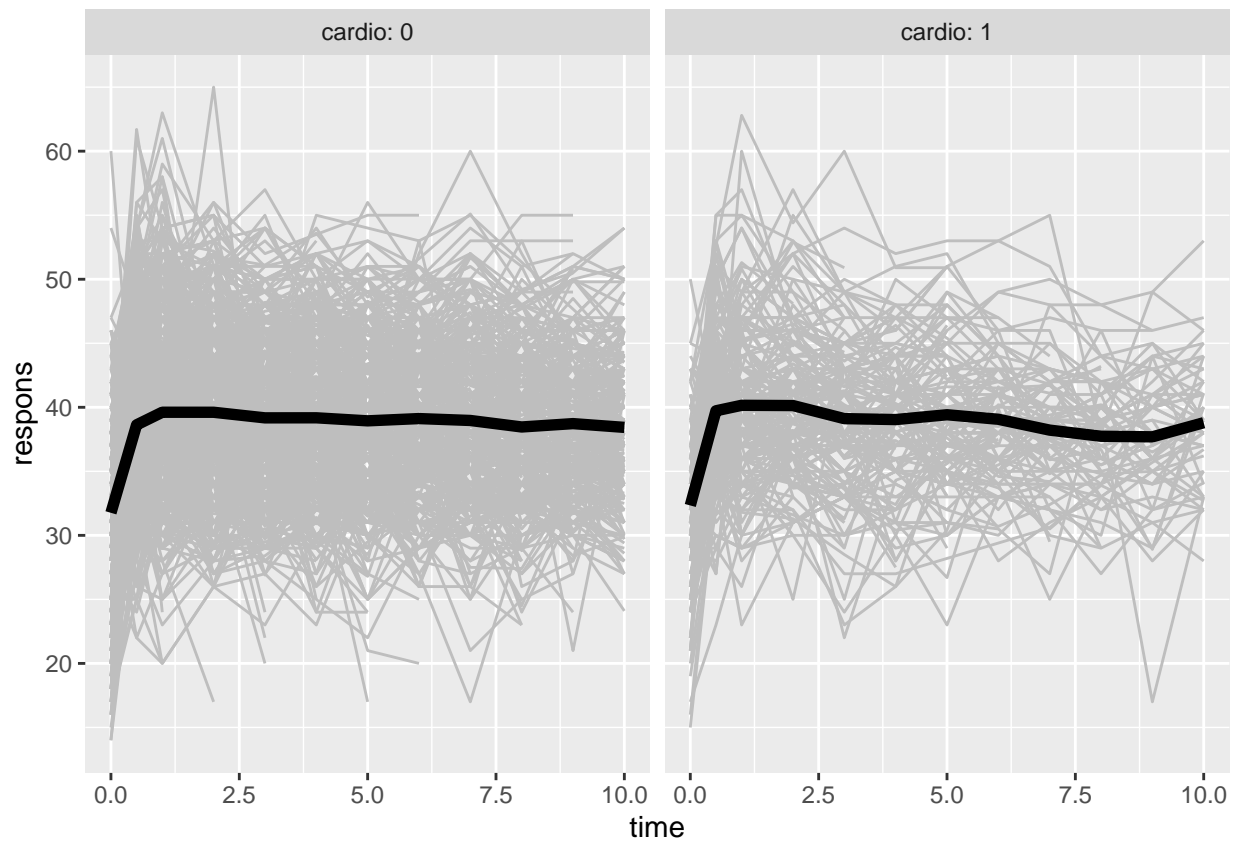


```
# Spaghetti Ggplot separated by male =1
p <- ggplot(data=data.long.noNA,aes(x=time,y=respons,group=id))
p <- p + geom_line(col="grey")+stat_summary(aes(group=1),geom="line",fun=mean,linewidth=2)
p + facet_grid(~male,labeller=label_both)
```

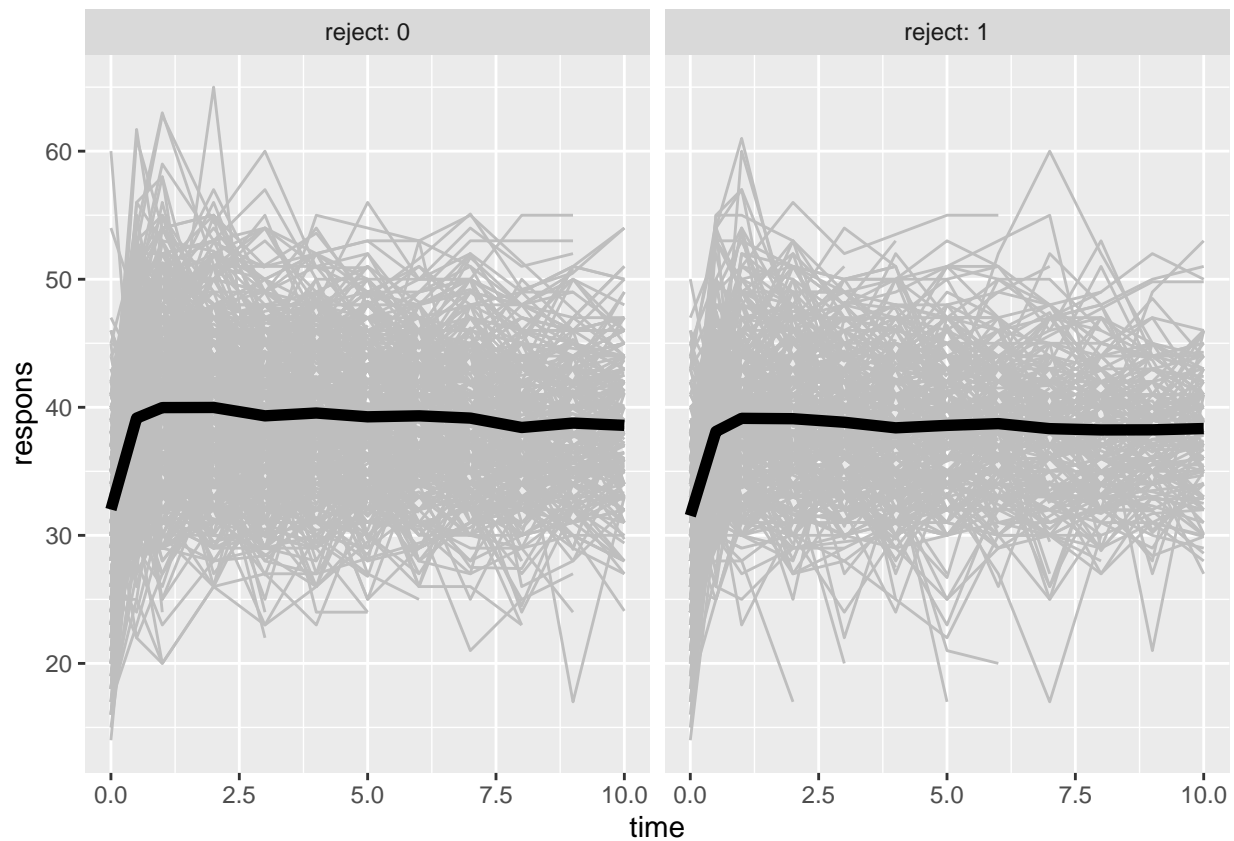




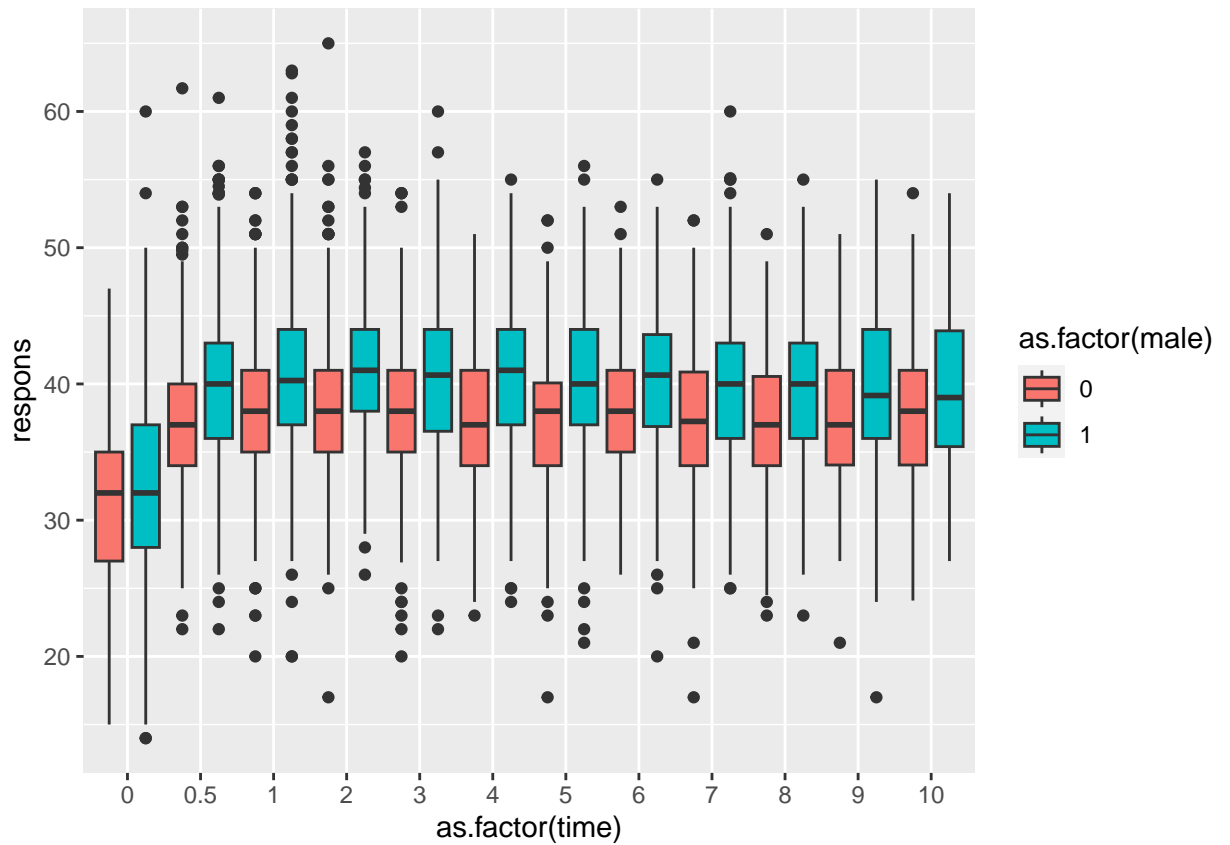
```
# Spaghetti Ggplot separated by cardio
p <- ggplot(data=data.long.noNA,aes(x=time,y=respons,group=id))
p <- p + geom_line(col="grey")+stat_summary(aes(group=1),geom="line",fun=mean,linewidth=2)
cardio.labs <- c("Cardio = 0","Cardio = 1")
p + facet_grid(~cardio,labeller = label_both)
```



```
# Spaghetti Ggplot separated by reject =1
p <- ggplot(data=data.long.noNA,aes(x=time,y=respons,group=id))
p <- p + geom_line(col="grey")+stat_summary(aes(group=1),geom="line",fun=mean,linewidth=2)
p + facet_grid(~reject,labeller=label_both)
```

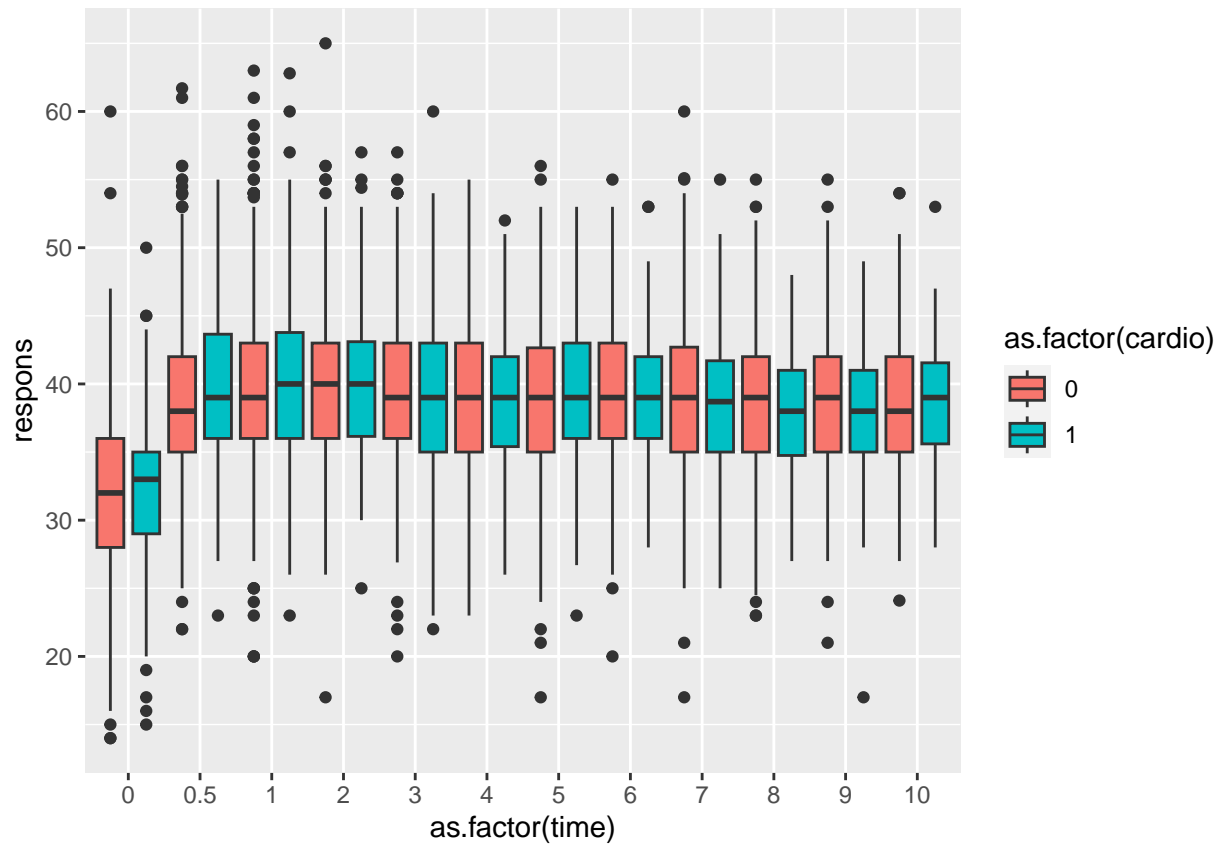


```
# Box plot by sex
ggplot(data.long.noNA,aes(x=as.factor(time),y=respons,fill=as.factor(male)))+
  geom_boxplot(position=position_dodge(1))
```

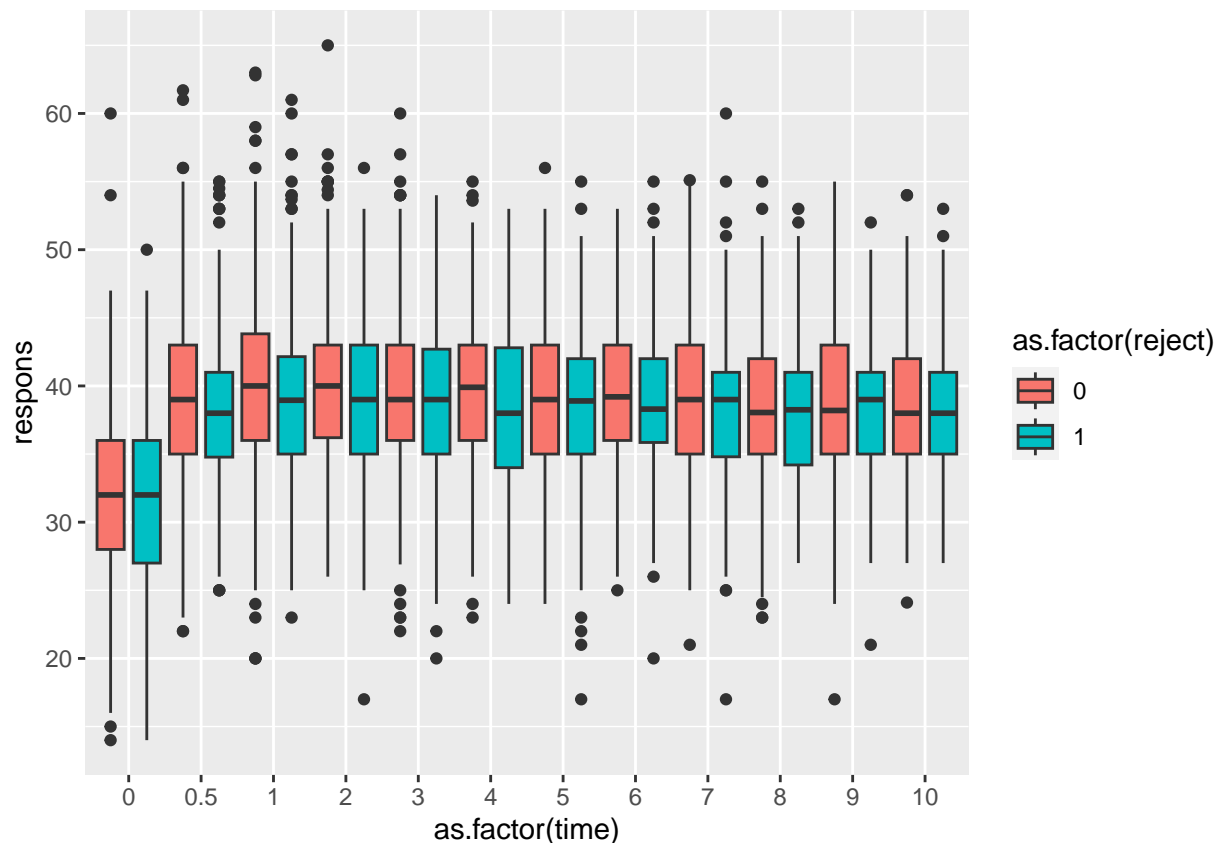


Boxplot

```
# Box plot by cardio
ggplot(data.long.noNA,aes(x=as.factor(time),y=respons,fill=as.factor(cardio)))+
  geom_boxplot(position=position_dodge(1))
```



```
# Box plot by reject
ggplot(data.long.noNA,aes(x=as.factor(time),y=respons,fill=as.factor(reject)))+
  geom_boxplot(position=position_dodge(1))
```



#### Correlation plot of Hc levels in different time

```
HcCorr = trenal.wide[,c(1:12)]
cor(HcCorr,use="complete.obs" ) # also COV for covariance
```

	HC0	HC06	HC1	HC2	HC3	HC4	HC5
## HC0	1.00000000	0.2264123	0.1587116	0.1724777	0.2139805	0.1732267	0.1557624
## HC06	0.22641235	1.0000000	0.7562367	0.6233688	0.5520591	0.5278499	0.5143061
## HC1	0.15871158	0.7562367	1.0000000	0.7315995	0.6656006	0.6119867	0.5873331
## HC2	0.17247771	0.6233688	0.7315995	1.0000000	0.7284046	0.6382434	0.5996189
## HC3	0.21398049	0.5520591	0.6656006	0.7284046	1.0000000	0.7733522	0.7016965
## HC4	0.17322666	0.5278499	0.6119867	0.6382434	0.7733522	1.0000000	0.7888249
## HC5	0.15576243	0.5143061	0.5873331	0.5996189	0.7016965	0.7888249	1.0000000
## HC6	0.13620085	0.4569881	0.5004036	0.4869519	0.5786122	0.6814132	0.7592203
## HC7	0.10156045	0.3936597	0.4541699	0.4724703	0.5402798	0.6466212	0.7067887
## HC8	0.08419757	0.3687935	0.4454882	0.4244221	0.5030428	0.6040136	0.6080051
## HC9	0.08859254	0.3711560	0.4254622	0.3971477	0.4303661	0.5461579	0.5713338
## HC10	0.09718506	0.4210917	0.4301937	0.4647890	0.4972001	0.5629570	0.5800544
	HC6	HC7	HC8	HC9	HC10		
## HC0	0.1362008	0.1015604	0.08419757	0.08859254	0.09718506		
## HC06	0.4569881	0.3936597	0.36879347	0.37115604	0.42109175		
## HC1	0.5004036	0.4541699	0.44548815	0.42546216	0.43019368		
## HC2	0.4869519	0.4724703	0.42442213	0.39714773	0.46478897		
## HC3	0.5786122	0.5402798	0.50304282	0.43036614	0.49720006		
## HC4	0.6814132	0.6466212	0.60401365	0.54615793	0.56295695		
## HC5	0.7592203	0.7067887	0.60800514	0.57133378	0.58005440		
## HC6	1.0000000	0.7414970	0.67347761	0.62938253	0.60329422		

```

## HC7  0.7414970 1.0000000 0.71838142 0.63933448 0.65646214
## HC8  0.6734776 0.7183814 1.00000000 0.70316750 0.68501304
## HC9  0.6293825 0.6393345 0.70316750 1.00000000 0.74259683
## HC10 0.6032942 0.6564621 0.68501304 0.74259683 1.00000000

library("PerformanceAnalytics")

## Warning: package 'PerformanceAnalytics' was built under R version 4.2.3
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
##
## ##### WARNING #####
## # We noticed you have dplyr installed. The dplyr lag() function breaks how      #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                              #
## # Calls to lag(my_xts) that you enter or source() into this session won't    #
## # work correctly.                                                            #
## #                                                                              #
## # All package code is unaffected because it is protected by the R namespace  #
## # mechanism.                                                                #
## #                                                                              #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop  #
## # dplyr from breaking base R's lag() function.                            #
## ##### WARNING #####
##
## Attaching package: 'xts'
## The following objects are masked from 'package:dplyr':
##
##      first, last
##
## Attaching package: 'PerformanceAnalytics'
## The following object is masked from 'package:graphics':
##
##      legend

chart.Correlation(HcCorr,historgram=TRUE)

## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter

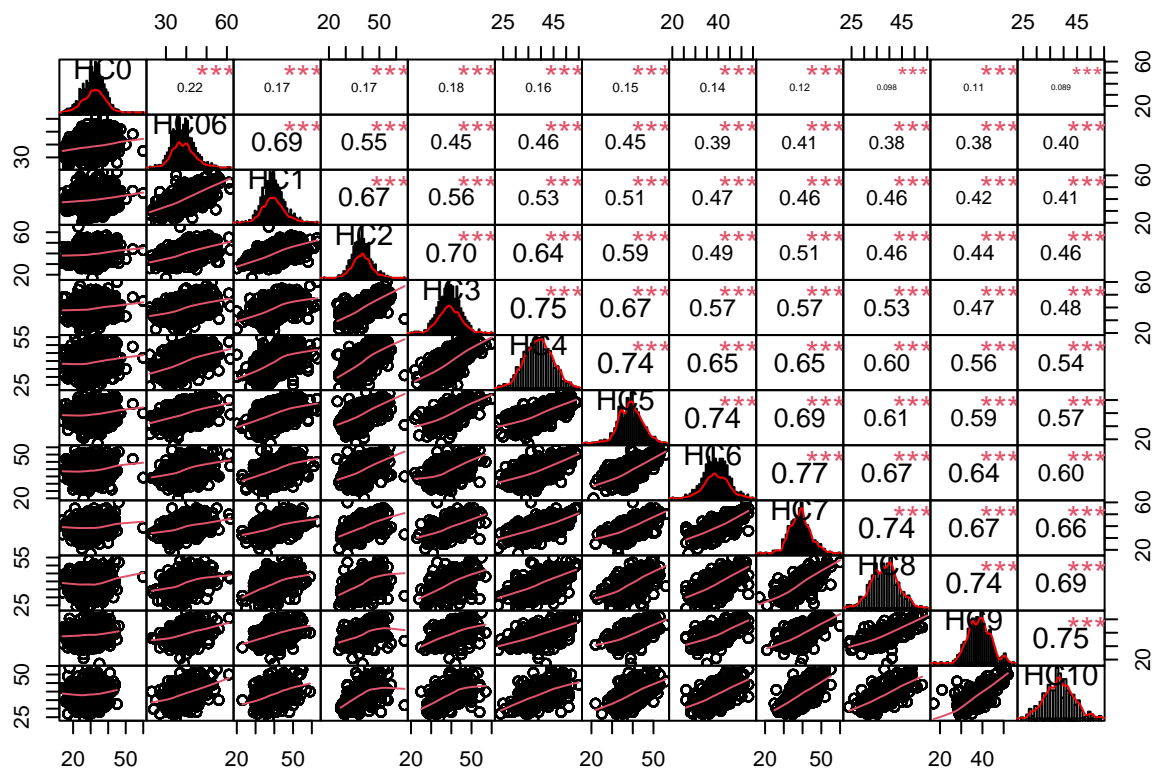
```

[illegible]



[illegible]

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```



Hypothesis based on the plot 1.Age 2.Male 3.Cardio 4.Reject

## Data set Trenal.XLS analysis with the linear mixed effects model

### The chosen of fixed effects variable

We can choose all the predictors as the fixed effect variables, plus an intercept ## The chosen of random effects variable

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

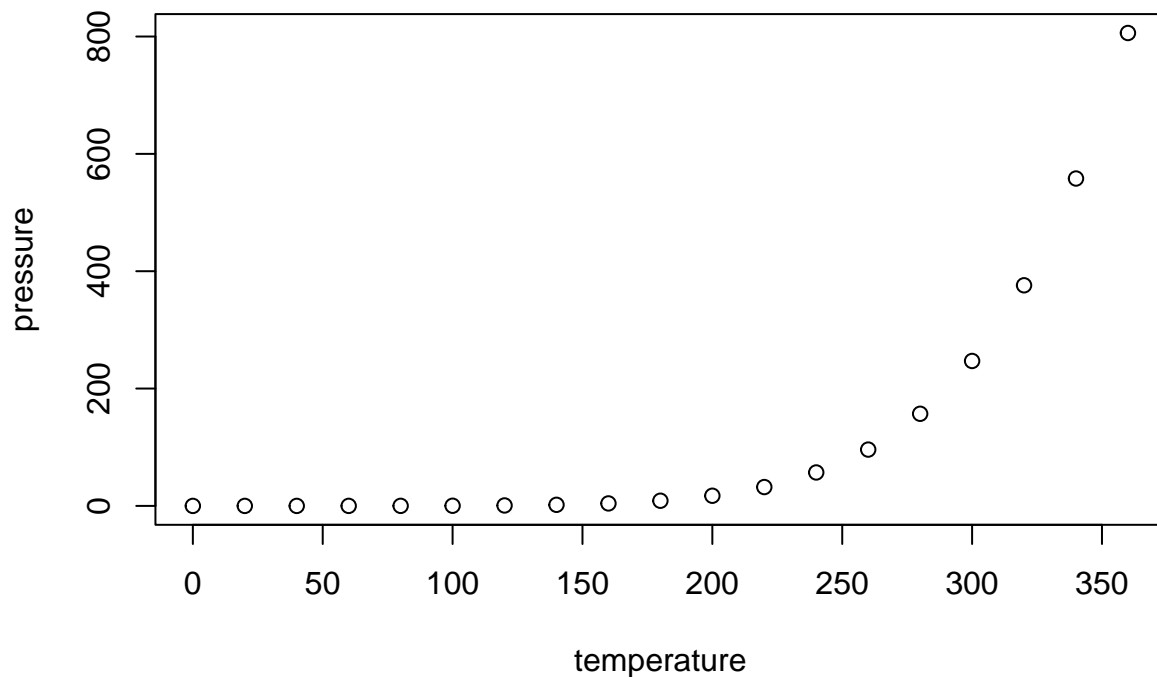
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##   Mean  :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max.  :25.0    Max.   :120.00
```

## Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.