Human stress responses in office-like environments with wood furniture: Data processing and analysis

Michael Burnard | InnoRenew CoE & University of Primorska |

5 June 2019

Contents

Introduction

This document contains the code and procedures for the analysis of salivary free cortisol levels used as an indicator of stress in “Human stress responses in office-like environments with wood furniture”.

The analysis includes extracting cortisol levels and other information from the microtitre plates saliva samples were processed on and various statistical, summary, and visual analyses used to understand the data and determine the results of the experiment.

The data are not included because individuals and their health data may be identifiable. Sanitized, psuedoanonymous data may be available upon request in qualified circumstances.

Acknowledgements

The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Slovenian infrastructure programme IO-0035.

Setting the Environment

```
knitr::opts_chunk$set(cache = TRUE)
library(drc)
library(tidyverse)
library(stringr)
library(ggforce)
library(scales)
library(zoo)
```

Data

Prepare storage data frames

Here we:

1. Create a data frames to store our important info
   a. Results (readings, cortisol levels, participant id, test info)
   b. Information about the model and its fit
   c. Information about each processed tray (calibrators, controls, etc)

2. Create a list to store models. We want to come back to inspect them later.

```
results <- tibble("Subject" = as.character(), "Test" = as.integer(),
                      "Interval" = as.numeric(), "Tray" = as.integer(),
                      "Reading.mu" = as.double(), "Cortisol.fitted.ng" = as.double(),
                      "Cortisol.fitted.nmol"=as.numeric())
control <- tibble("Name" = as.character(), "Reading.mu" = as.double(),
                      "Tray" = as.integer(), "Cortisol.fitted.ng" = as.double(),
                      "Cortisol.fitted.nmol"=as.numeric())
model.info <- tibble("Tray" = as.integer(), "b" = as.double(),
                         "c" = as.double(), "d" = as.double(),
                         "e" = as.double(), "Resid.var" = as.double())
tray.info <- tibble("Tray" = as.integer(), "Reading.name" = as.character(),
                        "Reading.value_1" = as.double(),
                        "Reading.value_2" = as.double(), "Reading.mu" = as.double(),
                        "Reading.absDiff"=as.double(),
                        "Fit.value.ng" = as.double(), "Fit.value.nmol"=as.numeric(),
                        "ExpectedRaw.value" = as.double(),
                        "ExpectedFit.value" = as.numeric())
models.list <- vector("list", 25)
```

Read, parse, and calculate cortisol concentrations

Data files from the microtitre plate reader were stored in a single folder that was read, processed, and stored as follows. Files associating the readings with subject identifiers were also read and matched to the data for processing.

```r
for(i in 1:25)
{
  #read data
  readings.path <- paste("data/cortisol/Readings/CortisolReadings_tray",
                         as.character(i), ".csv", sep="")
  layout.path <-  paste("data/cortisol/Layouts/CortisolLayout_tray",
                         as.character(i), ".csv", sep="")

  readings.all <- read.csv(readings.path, stringsAsFactors=FALSE)
  layout.all <- read.csv(layout.path, stringsAsFactors=FALSE)

  #prep data
  readings.all <- gather(readings.all, "Col", "Reading", 2:13)
  readings.all$Col <- as.numeric(str_sub(readings.all$Col, 2))
  layout.all <- gather(layout.all, "Col", "Name", 2:13)
  readings.all$Name <- layout.all$Name

  #Split data into useful groups
  readings.subject <- readings.all %>% filter(str_length(Name) > 5)
  readings.calib <- readings.all %>% filter(str_detect(Name, "C{1}[0123456]") == TRUE)
  readings.control_a <- readings.all %>% filter(str_detect(Name, "C{1}[LHM]") == TRUE)
  readings.control_b <- readings.all %>% filter(str_detect(Name, "C{1}[o]") == TRUE)

  #two temporary storage frames
  readings.calib_t <- readings.calib[,3:4]
  readings.control_t <- bind_rows(readings.control_a[,3:4], readings.control_b[,3:4])

  #finalise data prep
  readings.subject <- readings.subject %>%
    separate(Name, c("Subject", "Test", "Interval", "Position"), convert=TRUE) %>%
    group_by(Subject, Test, Interval) %>% summarise(Reading.mu = mean(Reading)) %>%
    mutate(Tray = i)
  readings.calib <- readings.calib %>% separate(Name, c("Name", "Position")) %>%
    group_by(Name) %>% summarise(Reading.mu = mean(Reading)) %>% mutate(Tray = i)
  readings.control_a <- readings.control_a %>% separate(Name, c("Name", "Position")) %>%
     group_by(Name) %>% summarise(Reading.mu = mean(Reading)) %>% mutate(Tray = i)
  readings.control_b <- readings.control_b %>% ungroup() %>%
    mutate(Reading.mu = Reading, Tray = i)
  readings.control <- bind_rows(readings.control_a, readings.control_b[,c(4,5,6)])
  #the same except for summarise for the temp storage frames
  readings.calib_t <- readings.calib_t %>% separate(Name, c("Name", "Position"))
  readings.control_t <- readings.control_t %>% separate(Name, c("Name", "Position"))

  #add cortisol values for model fitting & storage
  readings.calib$Cortisol <- c(0.0, 0.5, 1, 5, 10, 20, 100)

  #fit a model and store it in our list of models to look at later.
  #there is a special case for tray 23 because of a robot error
  #the robot missed a row (probably a pipette wasn't well attached)
  if(i == 23) {
    readings.calib <- readings.calib %>% filter(Name != "C1")
    m <- drm(Cortisol~Reading.mu, data=readings.calib, fct=LL.4())
    models.list[[i]] <- m
    names(models.list)[[i]] <- paste("Tray", i, sep="_")
    tmp.row <- data_frame("Name"="C1", "Reading.mu"=NA, "Tray"=i, "Cortisol"=0.5)
    readings.calib <- bind_rows(readings.calib, tmp.row) %>% arrange(Cortisol)
  } else {
    m <- drm(Cortisol~Reading.mu, data=readings.calib, fct=LL.4())
    models.list[[i]] <- m
    names(models.list)[[i]] <- paste("Tray", i, sep="_")
  }

  #store model info in a data frame to compare
  temp.row <- data.frame("Tray" = i, "b" = coef(m)[[1]], "c" = coef(m)[[2]],
                         "d" = coef(m)[[3]], "e" = coef(m)[[4]],
                         "Resid.var" = summary(m)[[1]])
  model.info <- bind_rows(model.info, temp.row)

  #store calibration and control readings
  readings.calib_t <- spread(readings.calib_t, Position, Reading)
  temp.df <- as.data.frame(cbind(rep(i, 7),
                         readings.calib_t$Name,
                         readings.calib_t[2],
                         readings.calib_t[3],
                         (readings.calib_t[2] + readings.calib_t[3]) / 2,
                         abs(readings.calib_t[2] - readings.calib_t[3]),
                         rep(1,7),
                         c(2.643, 2.229, 2.056, 1.295, 0.789, 0.451, 0.131),
                         readings.calib$Cortisol))
  names(temp.df) <- c("Tray", "Reading.name", "Reading.value_1" ,
                      "Reading.value_2", "Reading.mu",
                      "Reading.absDiff", "Fit.value.ng",
                      "ExpectedRaw.value", "ExpectedFit.value")
  temp.df$Fit.value.ng <- fitted(m, temp.df[,c(4,1)])
  temp.df$Fit.value.nmol <- temp.df$Fit.value.ng * 2.76 # add nmol/L value

  #another correction for tray 23
  if(i == 23) { temp.df <- temp.df[-2,]}
  tray.info <- bind_rows(tray.info, temp.df)
  tray.info$ExpectedFit.value.nmol <- tray.info$ExpectedFit.value * 2.76

  #use the fitted model to back calculate
  readings.subject$Cortisol.fitted.ng <- fitted(m, as.data.frame(readings.subject[,4]))
  readings.subject$Cortisol.fitted.nmol <- readings.subject$Cortisol.fitted.ng * 2.76
  results <- bind_rows(results, readings.subject[,c(1,2,3,5,7,4,6)])

  #and for controls
  readings.control$Cortisol.fitted.ng <- fitted(m, as.data.frame(readings.control[,2]))
  readings.control$Cortisol.fitted.nmol <- readings.control$Cortisol.fitted.ng * 2.76
  control <- bind_rows(control, readings.control)

}
```

Now, merge Test Record data (video, room, test sequence, etc.) with results. We also make an pseudoanonymous identifier that isn't associable with the subject identities. Because of anonymisation, this isn't run here. Instead we read a file with the saved results.

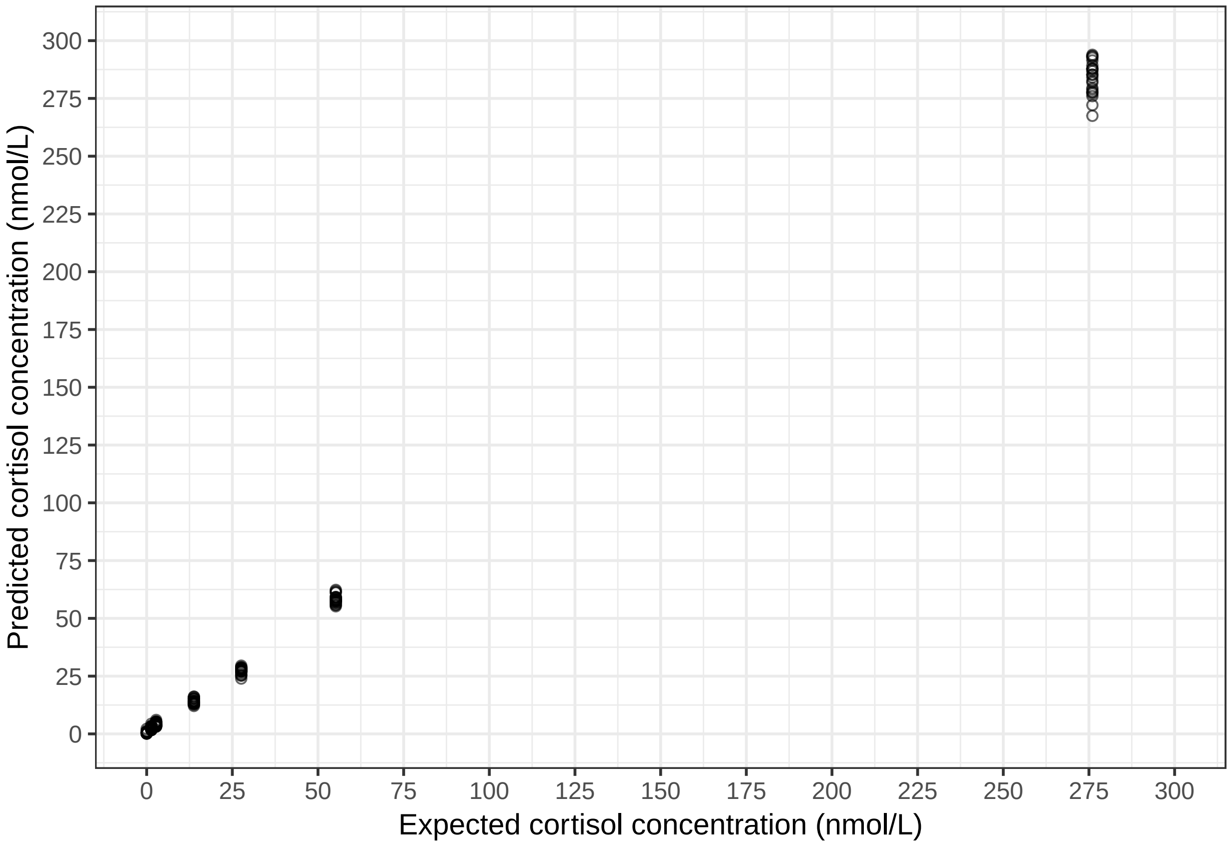
```
#Read the file and select the columns of interest
tr <- read.csv("data/TestRecord.csv", stringsAsFactors=FALSE)
tr <- tr %>% select(Participant, TestType, Room, Video, SGL, WHO5_Total)
#rename for the merge
names(tr)[1] <- "Subject"
names(tr)[5] <- "Test"

#Merge them with the results
results <- left_join(results, tr, by=c("Subject", "Test"))
```

```r
#Add a character string to further anonomise the results in output plots.
makeRandString <- function() {
  tmp = c(sample(LETTERS, 3, replace=TRUE),
    sample(0:9, 2, replace=TRUE))
  return(paste0(tmp, collapse=""))
}


results <- results %>% group_by(Subject) %>% mutate(Anon=makeRandString())
# verify unique for each subject
length(unique(results$Anon))
```

```r
results <- read.csv("results_storage.csv", stringsAsFactors = FALSE, header = TRUE)
#to double check we have 14 total for each subject

verify <- data.frame(table(results$Subject, results$Tray))
verify <- spread(verify, Var2, Freq)
verify <- verify %>% mutate(RowSum = rowSums(verify[,2:26]))

#The pipette robot had a small malfuction (didn't seal a pipette tip) on one tray
#rendering the values from row unuseable.
results$Cortisol.fitted.ng <- ifelse(results$Tray == 23 & results$Interval == 2, NA,
                                     results$Cortisol.fitted.ng)
results$Cortisol.fitted.nmol <- ifelse(results$Tray == 23 & results$Interval == 2, NA,
                                       results$Cortisol.fitted.nmol)

#add time in minutes
results$Minutes <- ifelse(results$Interval == 1, 0,
                   ifelse(results$Interval == 2, 15,
                   ifelse(results$Interval == 3, 25,
                   ifelse(results$Interval == 4, 35,
                   ifelse(results$Interval == 5, 45,
                   ifelse(results$Interval == 6, 60, 75))))))

#3 subject-test combos were duplicated on tray 25 for verification
# XX-2, XX-1, XX-2. Need to remove these from the general results
#and keep them to check.
check <- results %>% filter ((Anon %in% c("XGD55","FQX13","KH049") & Tray == 25))
results <- results %>% filter( !(Anon %in% c("XGD55","FQX13","KH049") & Tray == 25) )
```

#Plate to Plate comparisons and results

```r
ggplot(data=tray.info,
       aes(x=ExpectedFit.value * 2.76, y=Fit.value.ng * 2.76)) + theme_bw() +
  geom_point(alpha=.6, shape=1) +
  scale_x_continuous(limits=c(0,300), breaks=seq(0,300,25)) +
  scale_y_continuous(limits=c(0,300), breaks=seq(0,300,25)) +
  labs(x="Expected cortisol concentration (nmol/L)",
       y="Predicted cortisol concentration (nmol/L)")
```



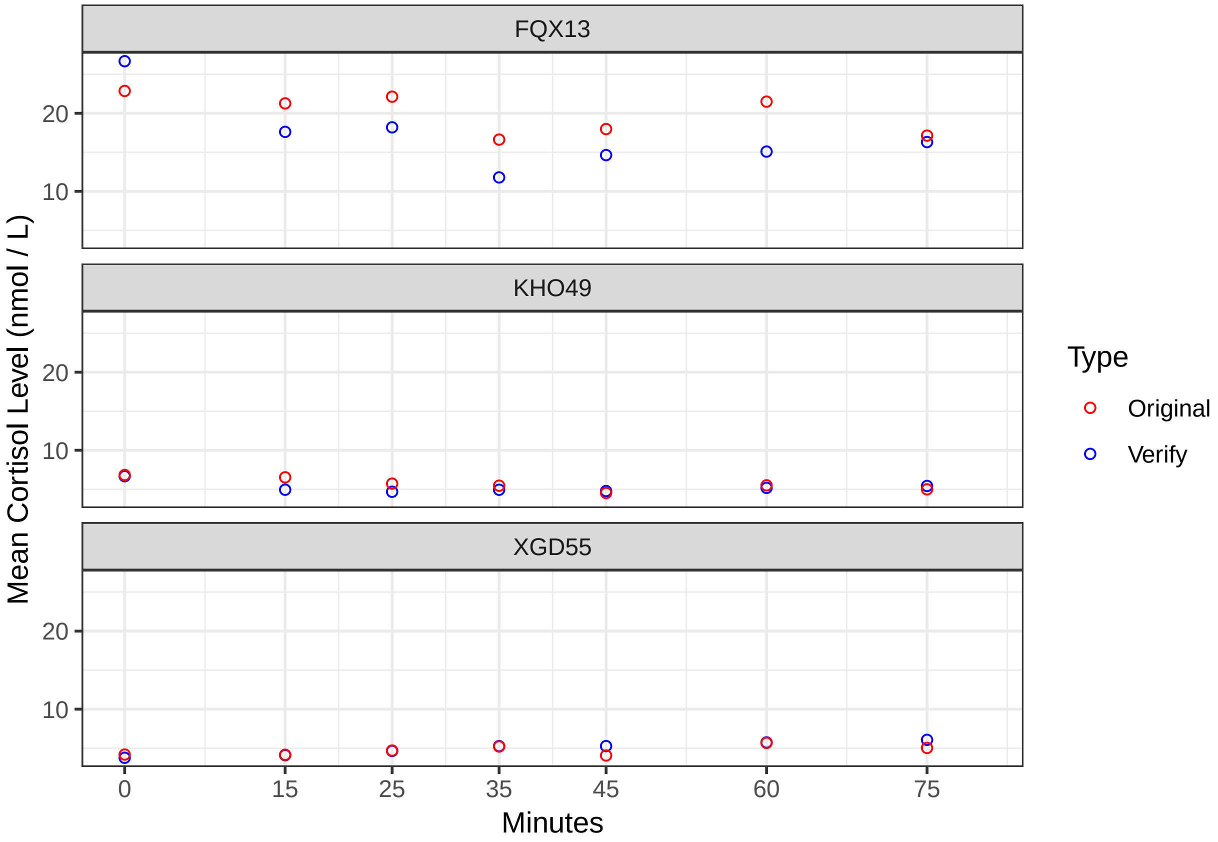
```
tray.fit <- tray.info %>% select(Tray, Fit.value.nmol, ExpectedFit.value.nmol) %>%
  group_by(Tray) %>%
  summarise(RMSE = sqrt(mean((Fit.value.nmol - ExpectedFit.value.nmol)^2)))

model.info$RMSE <- tray.fit$RMSE
#write.csv(file="Model_info.csv", model.info)
#write.csv(file="Calibrators.csv", tray.info %>% group_by(Reading.name)
#%>% summarise(mu = mean(Fit.value.nmol), med=median(Fit.value.nmol),
#sd=sd(Fit.value.nmol), coef.v=sd/mu, min=min(Fit.value.nmol),max=max(Fit.value.nmol)) )

#these are articially high, because of what we are measuring
#the results are near perfect because we are measuring curves fit to a known value
#to other curves fit to the same known value.
r2 <- 1 - (sum((tray.info$ExpectedFit.value.nmol - tray.info$Fit.value.nmol)^2) /
             sum((tray.info$Fit.value.nmol - mean(tray.info$Fit.value.nmol))^2))

rho <- cov(tray.info$Fit.value.nmol, tray.info$ExpectedFit.value.nmol) /
  (sd(tray.info$Fit.value.nmol) * sd(tray.info$ExpectedFit.value.nmol))

tray.info %>% group_by(Reading.name) %>% summarise(min=min(Reading.mu))
```

Reading.name	min
C0	1.7185
C1	1.4730
C2	1.2960
C3	0.8275
C4	0.5870
C5	0.3860
C6	0.1415

Duplicate testing

Three sample groups were tested in duplicate (on different trays) to guage tray-to-tray variation.

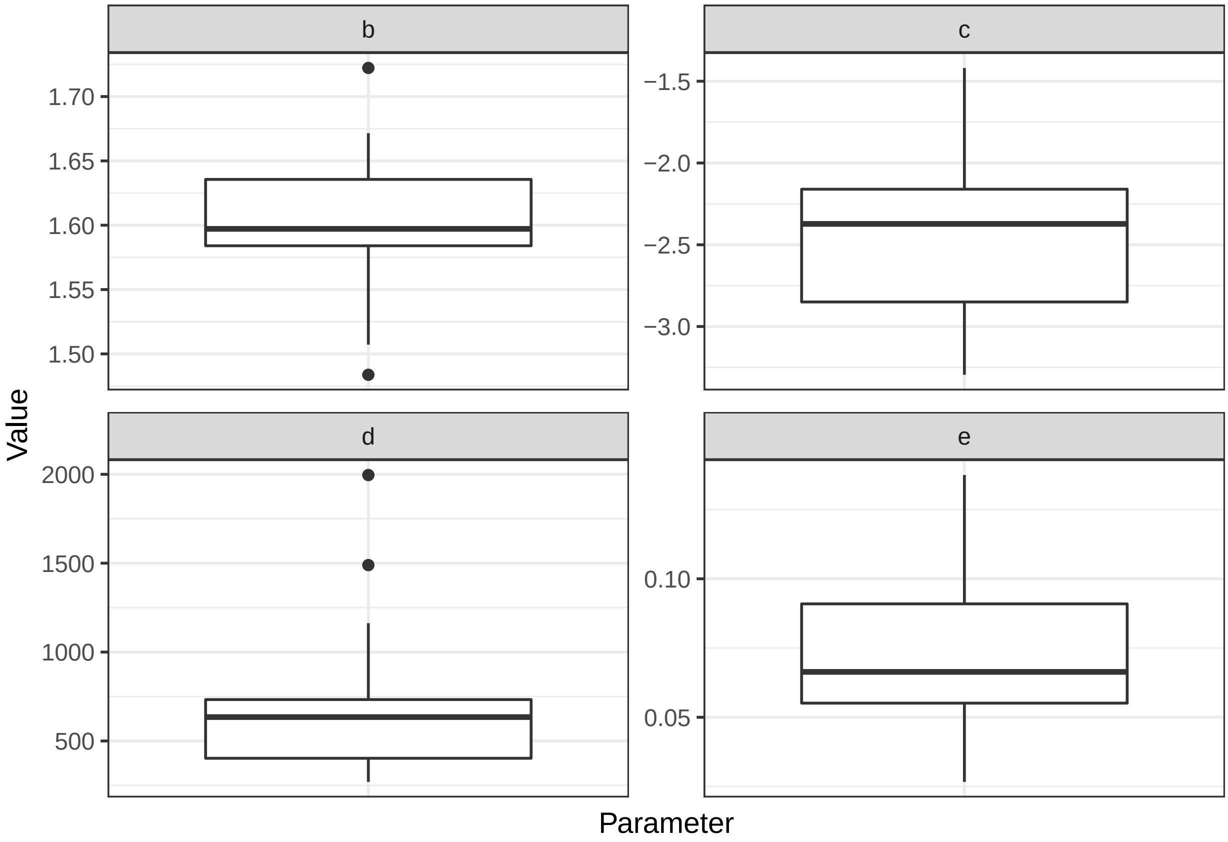
```
#plate to plate

#first, the duplicated trio

chck <- check %>% select(Subject, Anon, Test, Tray, Minutes, Reading.mu,
                         Cortisol.fitted.nmol)
r.chck <- results %>% select(Subject, Anon, Test, Tray, Minutes, Reading.mu,
                             Cortisol.fitted.nmol) %>%
  filter( (Anon == "XGD55" & Test == 1) |
          (Anon == "FQX13" & Test == 2) |
          (Anon == "KHO49" & Test == 2))
r.chck$Type <- "Original"
chck$Type <- "Verify"
chck <- bind_rows(chck, r.chck)

ggplot(data=chck, aes(x=Minutes, y=Cortisol.fitted.nmol, colour=Type)) + theme_bw() +
  geom_point(shape=1) +
  facet_wrap(~Anon, nrow=3) +
  scale_x_continuous(limits=c(0,80), breaks=c(0,15,25,35,45,60,75)) +
  scale_colour_manual(values=c("#FF0000", "#0000FF")) +
  labs(x="Minutes", y="Mean Cortisol Level (nmol / L)")
```



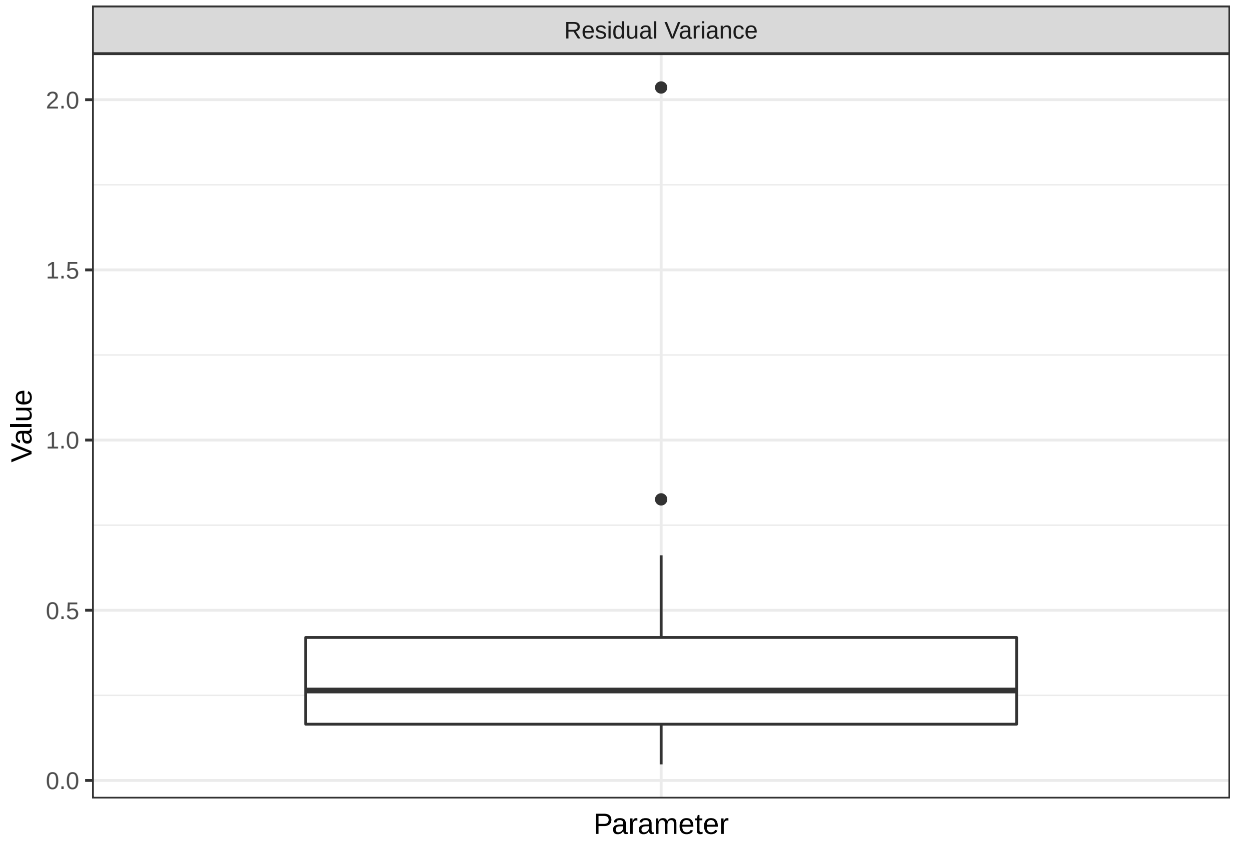
```
#now values
chck.o <- chck %>% filter(Type == "Original") %>%
  select(Subject, Cortisol.fitted.nmol)%>%  arrange(Subject)
chck.v <- chck %>% filter(Type == "Verify") %>%
  select(Subject, Cortisol.fitted.nmol) %>%  arrange(Subject)
names(chck.o)[2] <- "Original"
names(chck.v)[2] <- "Verify"

chck.o$Verify <- chck.v$Verify
chck.o$RawDiff <- abs(chck.o$Verify - chck.o$Original)
chck.o$PerDiff <- abs((chck.o$Verify - chck.o$Original) / chck.o$Verify)

chck.o <- chck.o %>% group_by(Subject) %>%
  summarise(raw.mu = mean(RawDiff), per.mu = mean(PerDiff))
```

Model and fit parameters

```
#consistency of fit parameters
mod.con <- gather(model.info, "Parameter", "Value", 2:6)
mod.con$Parameter <- ifelse(mod.con$Parameter == "Resid.var",
                           "Residual Variance", mod.con$Parameter)

ggplot(data=mod.con[which(mod.con$Parameter != "Residual Variance"),],
       aes(x=Parameter, y=Value)) + theme_bw() +
  geom_boxplot() +
  facet_wrap(~Parameter, scales="free") +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())
```



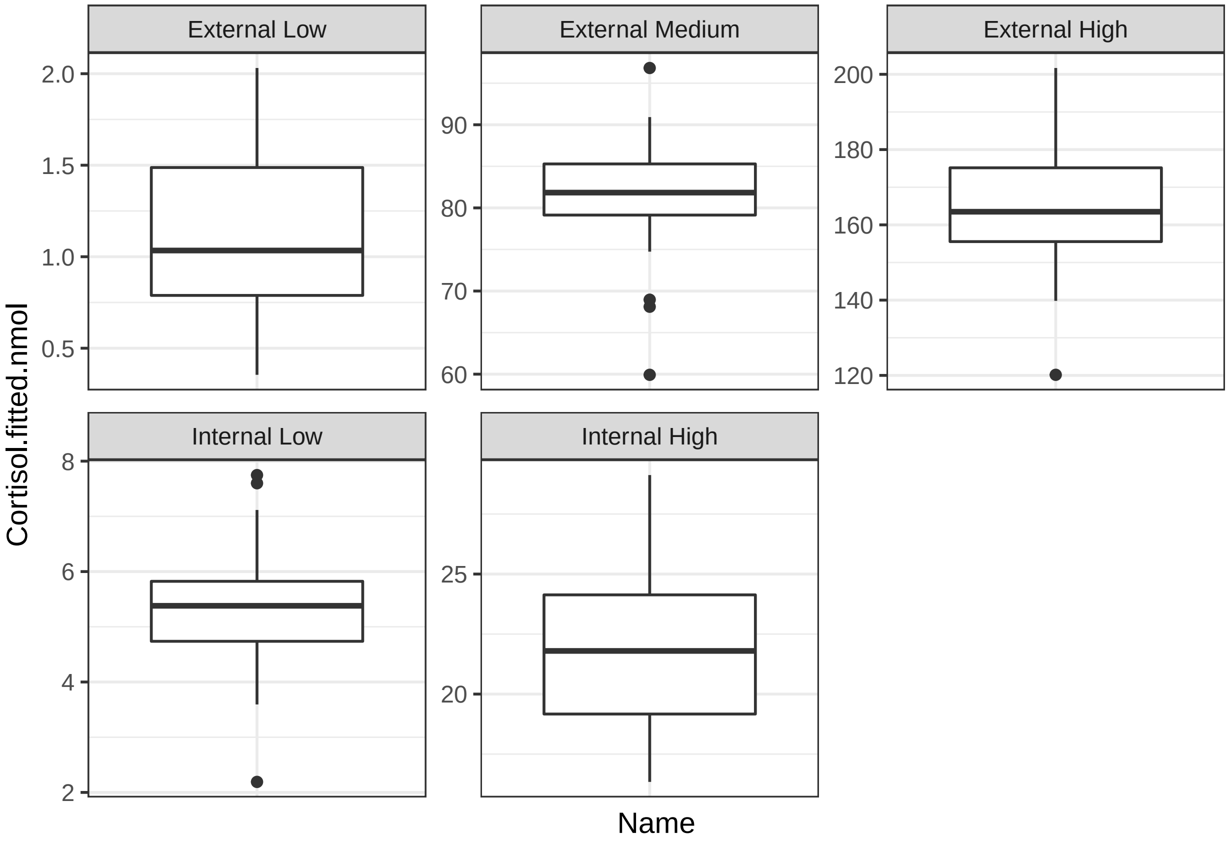
```
ggplot(data=mod.con[which(mod.con$Parameter == "Residual Variance"),],
       aes(x=Parameter, y=Value)) + theme_bw() +
  geom_boxplot() +
  facet_wrap(~Parameter) +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())
```



```
#plot(models.list$Tray_23)
```

Tray Controls

```
#Clear names for labels
control$Labels <- ifelse(control$Name == "CH", "External High",
                  ifelse(control$Name == "CL", "External Low",
                  ifelse(control$Name == "CM", "External Medium",
                  ifelse(control$Name == "Con-B", "Internal High",
                  ifelse(control$Name == "Con-A", "Internal Low", "Zed")))))
control$Labels <- as.factor(control$Labels)
control$Labels <- factor(control$Labels, levels=c("External Low", "External Medium",
                                                  "External High", "Internal Low",
                                                  "Internal High"))

ggplot(data=control %>% filter(Tray != 23), aes(x=Name, y=Cortisol.fitted.nmol)) +
  theme_bw() +
  geom_boxplot() +
  facet_wrap(~Labels, scales="free") +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank())
```

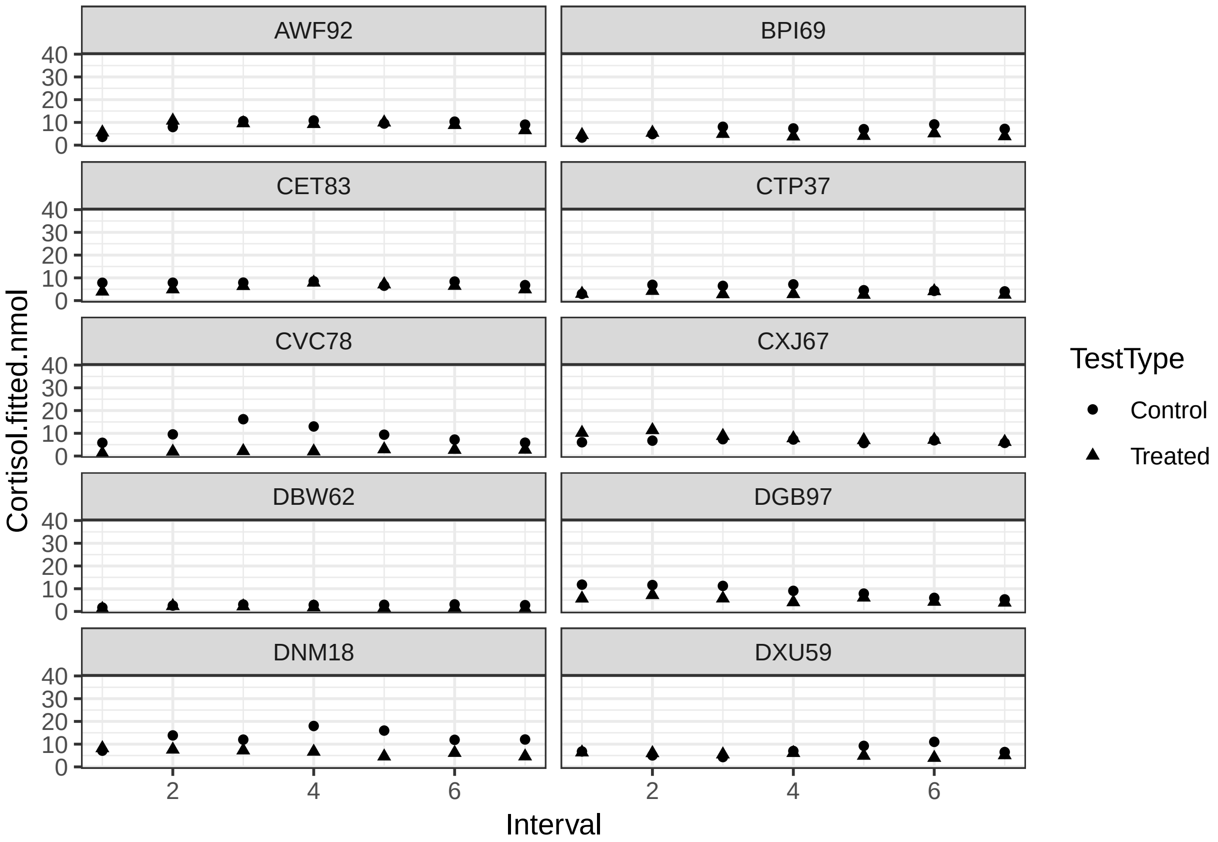


Results inspection for comparisons

Now we can graphically and statistically inspect the results.

```
#change page=X to see different pages.
ggplot(data=results, aes(x=Interval, y=Cortisol.fitted.nmol, shape=TestType)) +
  theme_bw() +
  geom_point() +
  facet_wrap_paginate(~Anon, ncol=2, nrow=5, page=1)
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

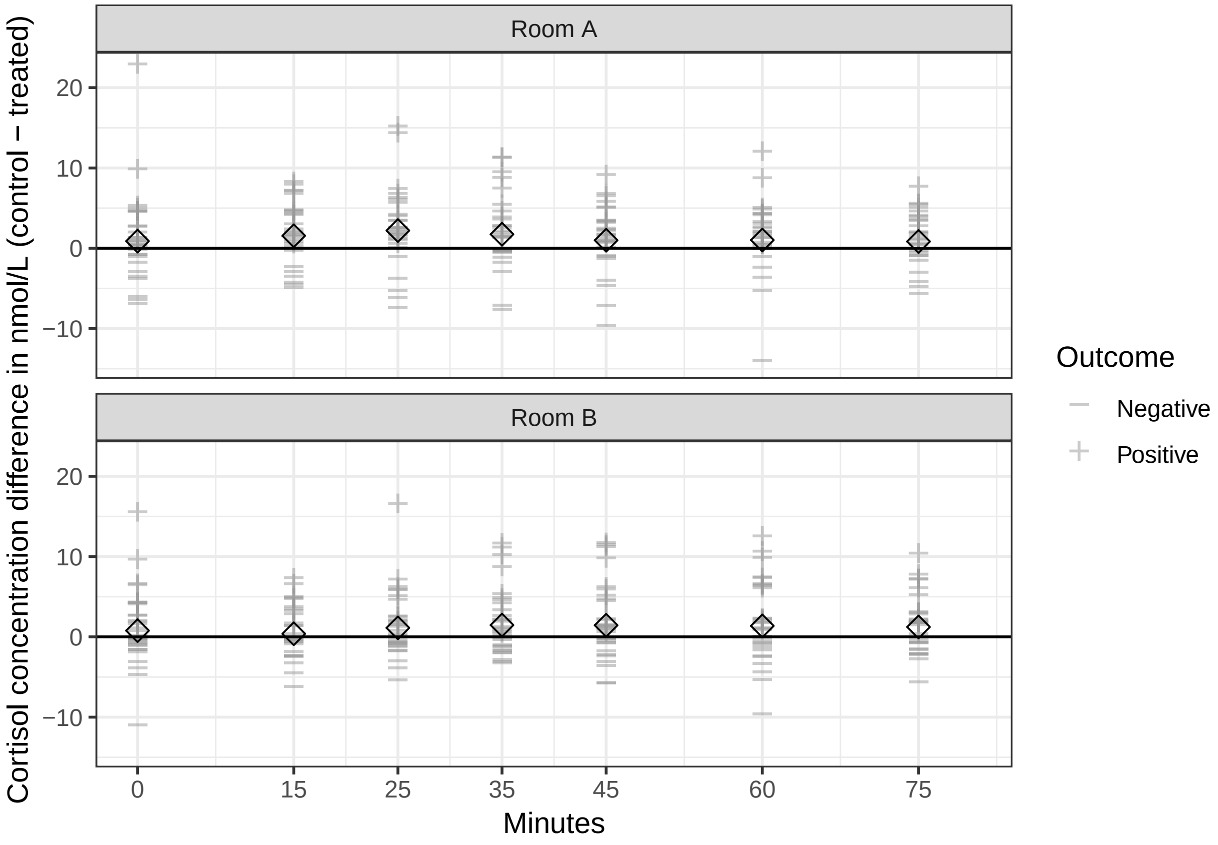


```
#generally all are fine except for DYG49. It's a very bizarre pattern for the
#treated test. Strange enough to warrant removal.
#The pattern doesn't follow expectations, and the values for tests are too
#different to provide much confidence. One situation or the other might be acceptable
#both are not.

r1 <- results %>% filter(Anon != "DYG49")

#first, check the responses to videos were not different


r1.vid <- r1 %>% group_by(Anon, Video) %>% filter(Interval %in% c(5,6,7)) %>%
  summarise(Test.mu=mean(Cortisol.fitted.nmol, na.rm=TRUE))
r1.vid <- r1.vid %>% group_by(Anon) %>% mutate(testDif=Test.mu - lag(Test.mu))
r1.vid$Outcome <- ifelse(is.na(r1.vid$testDif), "X",
                  ifelse(r1.vid$testDif < 0, "Positive",
                   "Negative"))
r1.vid$Outcome <- ifelse(r1.vid$Outcome == "X", lead(r1.vid$Outcome), r1.vid$Outcome)
r1.vid <- r1.vid %>% select(Anon, Video, Test.mu) %>%
  ungroup() %>% spread(Video, Test.mu)

wilcox.test(r1.vid$A, r1.vid$B, paired=TRUE)
```

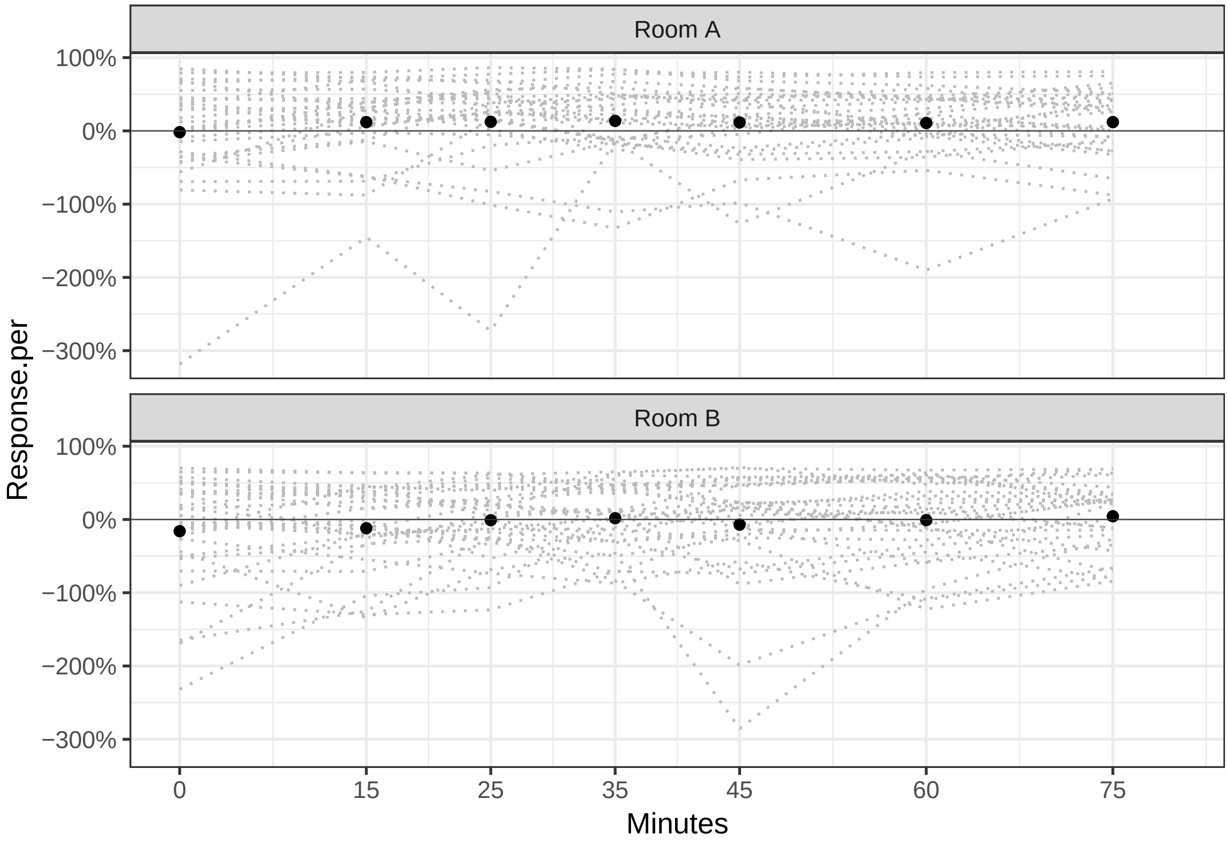
```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  r1.vid$A and r1.vid$B
## V = 881, p-value = 0.8052
## alternative hypothesis: true location shift is not equal to 0
```

```
r1.int <- r1 %>% select(Subject, Anon, Room, TestType, Interval,
                         Minutes, Cortisol.fitted.nmol) %>%
  spread("TestType", Cortisol.fitted.nmol) %>%
  mutate(Labels = ifelse(Room == "Oak", "Room A", "Room B"))
```

```r
r1.int <- r1.int[complete.cases(r1.int),]

r1.int$Response <- r1.int$Control - r1.int$Treated
r1.int$Response.per <- r1.int$Response / r1.int$Control
r1.int$Outcome <- ifelse(r1.int$Response > 0, "Positive", "Negative")
r1.int.mu <- r1.int %>% group_by(Labels, Minutes) %>%
  summarise(mu.raw = mean(Response),mu.per = mean(Response.per))
r1.int.mu$Outcome <- ifelse(r1.int.mu$mu.raw > 0, "Positive", "Negative")


ggplot(data=r1.int, aes(x=Minutes, y=Response, shape=Outcome)) + theme_bw() +
  geom_point(colour="#999999", alpha=0.5, size=5) +
  geom_point(data=r1.int.mu, aes(x=Minutes, y=mu.raw), shape=5, size=2.5) +
  geom_hline(yintercept=0, colour="#000000") +
  facet_wrap(~Labels, nrow=2) +
  scale_shape_manual(values=c(45,43)) +
  scale_x_continuous(limits=c(0,80), breaks=c(0,15,25,35,45,60,75)) +
  labs(y="Cortisol concentration difference in nmol/L (control - treated)")
```



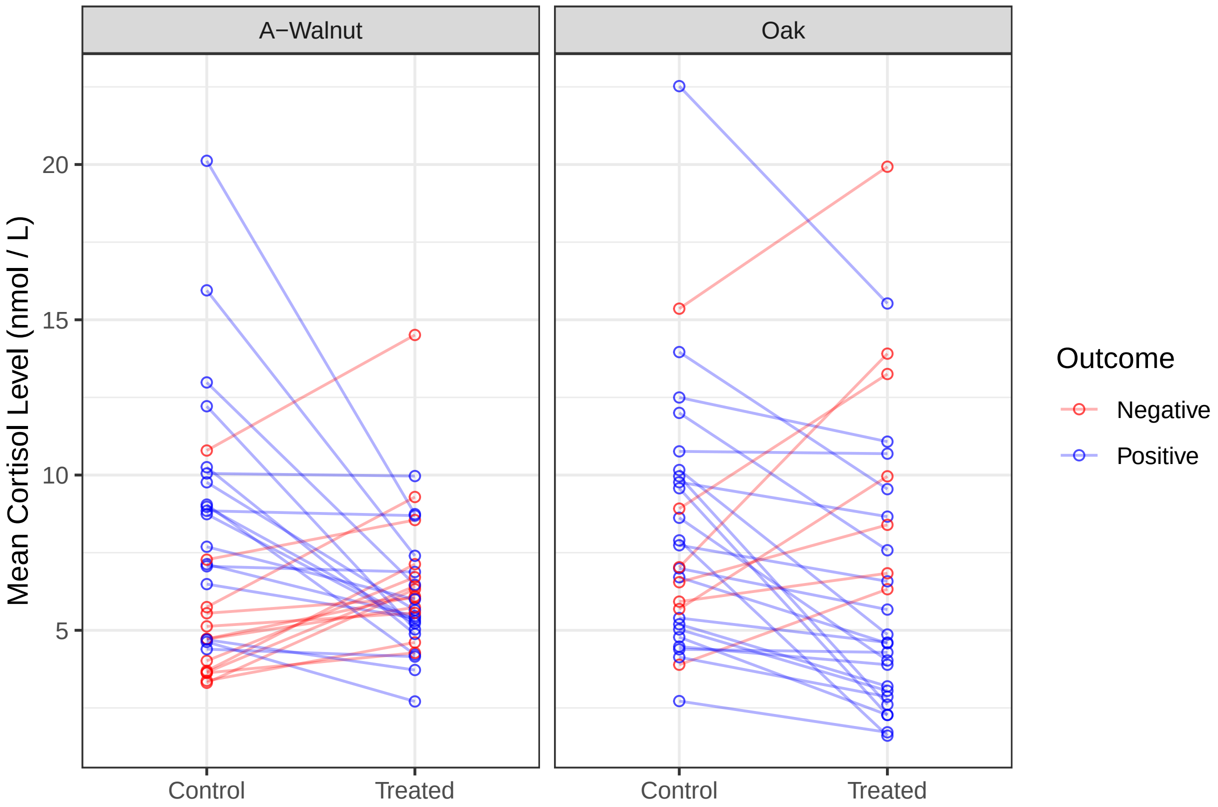
```r
#percent basis? Its difficult to interpret but offers another perspective
ggplot(data=r1.int, aes(x=Minutes, y=Response.per, group=Anon)) + theme_bw() +
  geom_line(colour="#BBBBBB", linetype=3) +
  geom_point(data=r1.int.mu, aes(x=Minutes, y=mu.per, group=1)) +
  geom_hline(yintercept=0, colour="#333333", size=0.2) +
  facet_wrap(~Labels, nrow=2) +
  scale_x_continuous(limits=c(0,80), breaks=c(0,15,25,35,45,60,75)) +
  scale_y_continuous(labels=percent_format())
```



Mean overall stress response

This is parametersied as the mean cortisol level in the control room minus the mean cortisol level in the treated room.

```
#overall mean per subject

####
# Check mean of interval-interval differences (should be the same)
####

r1.mu <- r1 %>% group_by(Subject, Anon, TestType, Room) %>%
  summarise(Test.mu=mean(Cortisol.fitted.nmol, na.rm=TRUE))

r1.mu <- r1.mu %>% group_by(Subject, Anon) %>% mutate(testDif=Test.mu - lag(Test.mu))
r1.mu$Outcome <- ifelse(is.na(r1.mu$testDif), "X",
                 ifelse(r1.mu$testDif < 0, "Positive",
                  "Negative"))
r1.mu$Outcome <- ifelse(r1.mu$Outcome == "X", lead(r1.mu$Outcome), r1.mu$Outcome)

r1.mu$Label <- ifelse(r1.mu$Room == "A-Walnut", "Walnut", "Oak")

ggplot(data=r1.mu, aes(x=TestType, y=Test.mu, colour=Outcome, group=Subject)) +
  theme_bw() +
  geom_line(alpha=.3) +
  geom_point(alpha=.7, shape=1) +
  facet_wrap(~Room) +
  scale_colour_manual(values=c("#FF0000", "#0000FF")) +
  labs(x="", y="Mean Cortisol Level (nmol / L)")
```



```
#create seperate data frames for each room
r1.mu.oak <- r1.mu %>% filter(Room == "Oak") %>%
  select(Subject, Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

r1.mu.wal <- r1.mu %>% filter(Room == "A-Walnut") %>%
  select(Subject, Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

#hist(r1.mu.wal$Control)
#hist(r1.mu.wal$Treated)
#hist(r1.mu.oak$Control)
#hist(r1.mu.oak$Treated)

#the data aren't normal enough, even with a log transform
#We use Wilcoxon rank sum, we're within-subjects so it's a paired test
#and with our parameterisation our alternative in this case is greater.
wilcox.test(r1.mu.oak$Control, r1.mu.oak$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
##
## data:  r1.mu.oak$Control and r1.mu.oak$Treated
## V = 317, p-value = 0.0154
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  0.2554169       Inf
## sample estimates:
## (pseudo)median
##       1.332559
```

```
wilcox.test(r1.mu.wal$Control, r1.mu.wal$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
##
```

```
##  Wilcoxon signed rank test
##
## data:  r1.mu.wal$Control and r1.mu.wal$Treated
## V = 313, p-value = 0.1047
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.231045       Inf
## sample estimates:
## (pseudo)median
##      0.8515471
```

```
#Summary info
r1.mu %>% filter(!is.na(testDif)) %>% group_by(Room, Outcome) %>%
  mutate(realtd=testDif*-1) %>% summarise(mu=mean(realtd), sd=sd(realtd),
                                        min=min(realtd), max=max(realtd), n=n())
```

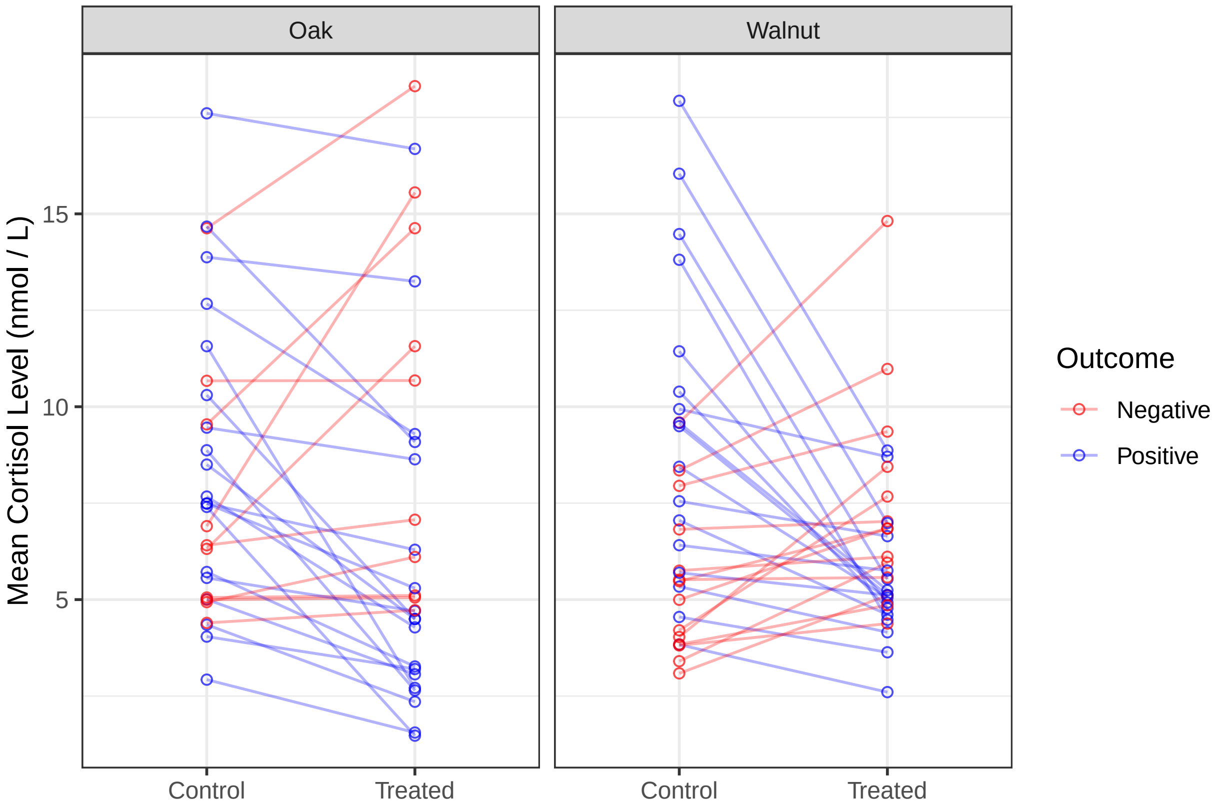
Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-1.973309	1.243451	-3.7216721	-0.4335732	13
A-Walnut	Positive	3.508476	3.269236	0.0844159	11.3775487	18
Oak	Negative	-3.607153	2.013998	-6.8777358	-0.9121788	7
Oak	Positive	2.914948	2.448286	0.0747172	7.3513351	22

```
r1.mu.per <- bind_rows(r1.mu.oak, r1.mu.wal)
r1.mu.per$Response <- r1.mu.per$Control - r1.mu.per$Treated
r1.mu.per$Response.per <- (r1.mu.per$Control - r1.mu.per$Treated) / r1.mu.per$Control
r1.mu.per$Outcome <- ifelse(r1.mu.per$Response > 0, "Positive", "Negative")
r1.mu.per %>% group_by(Room) %>% summarise(mu=mean(Response.per),
                                         sd=sd(Response.per), min=min(Response.per),
                                         max=max(Response.per), n=n())
```

Room	mu	sd	min	max	n
A-Walnut	0.0065084	0.4541373	-0.9305624	0.5883899	31
Oak	0.1359700	0.4426542	-0.9781982	0.7964564	29

Response period stress level

```
#mean per subject for periods intervals 4,5,6,7

r1.re <- r1 %>% group_by(Anon, TestType, Room) %>% filter(Interval %in% c(4,5,6,7)) %>%
  summarise(Test.mu=mean(Cortisol.fitted.nmol, na.rm=TRUE))

r1.re <- r1.re %>% group_by(Anon) %>% mutate(testDif=Test.mu - lag(Test.mu))
r1.re$Outcome <- ifelse(is.na(r1.re$testDif), "X",
                 ifelse(r1.re$testDif < 0, "Positive",
                  "Negative"))
r1.re$Outcome <- ifelse(r1.re$Outcome == "X", lead(r1.re$Outcome), r1.re$Outcome)
r1.re$Label <- ifelse(r1.re$Room == "A-Walnut", "Walnut", "Oak")
ggplot(data=r1.re, aes(x=TestType, y=Test.mu, colour=Outcome, group=Anon)) +
  theme_bw() +
  geom_line(alpha=.3) +
  geom_point(alpha=.7, shape=1) +
  facet_wrap(~Label) +
  scale_colour_manual(values=c("#FF0000", "#0000FF")) +
  labs(x="", y="Mean Cortisol Level (nmol / L)")
```



```
#create seperate data frames for each room
r1.re.oak <- r1.re %>% filter(Room == "Oak") %>%
  select(Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

r1.re.wal <- r1.re %>% filter(Room == "A-Walnut") %>%
  select(Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

#hist(r1.re.wal$Control)
#hist(r1.re.wal$Treated)
#hist(r1.re.oak$Control)
#hist(r1.re.oak$Treated)
#the data aren't normal enough, even with a log transform
#We use Wilcoxon rank sum, we're within-subjects so it's a paired test
#and with our parameterisation our alternative in this case is greater.
wilcox.test(r1.re.oak$Control, r1.re.oak$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
##
## data:  r1.re.oak$Control and r1.re.oak$Treated
## V = 315, p-value = 0.01727
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  0.3374182       Inf
## sample estimates:
## (pseudo)median
##       1.195808
```

```
wilcox.test(r1.re.wal$Control, r1.re.wal$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
```

```
##
## data:  r1.re.wal$Control and r1.re.wal$Treated
## V = 312, p-value = 0.1083
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.2290054        Inf
## sample estimates:
## (pseudo)median
##      0.9411696
```

```
r1.re %>% filter(!is.na(testDif)) %>% group_by(Room, Outcome) %>%
  mutate(realtd=testDif*-1) %>%
  summarise(mu=mean(realtd), sd=sd(realtd), min=min(realtd), max=max(realtd), n=n())
```

Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-1.937551	1.580826	-5.2227070	-0.0588927	14
A-Walnut	Positive	4.105112	3.364521	0.5815726	9.3348801	17
Oak	Negative	-2.495458	3.009454	-8.6497666	-0.0079062	10
Oak	Positive	3.073132	2.380199	0.6264034	8.8591524	19

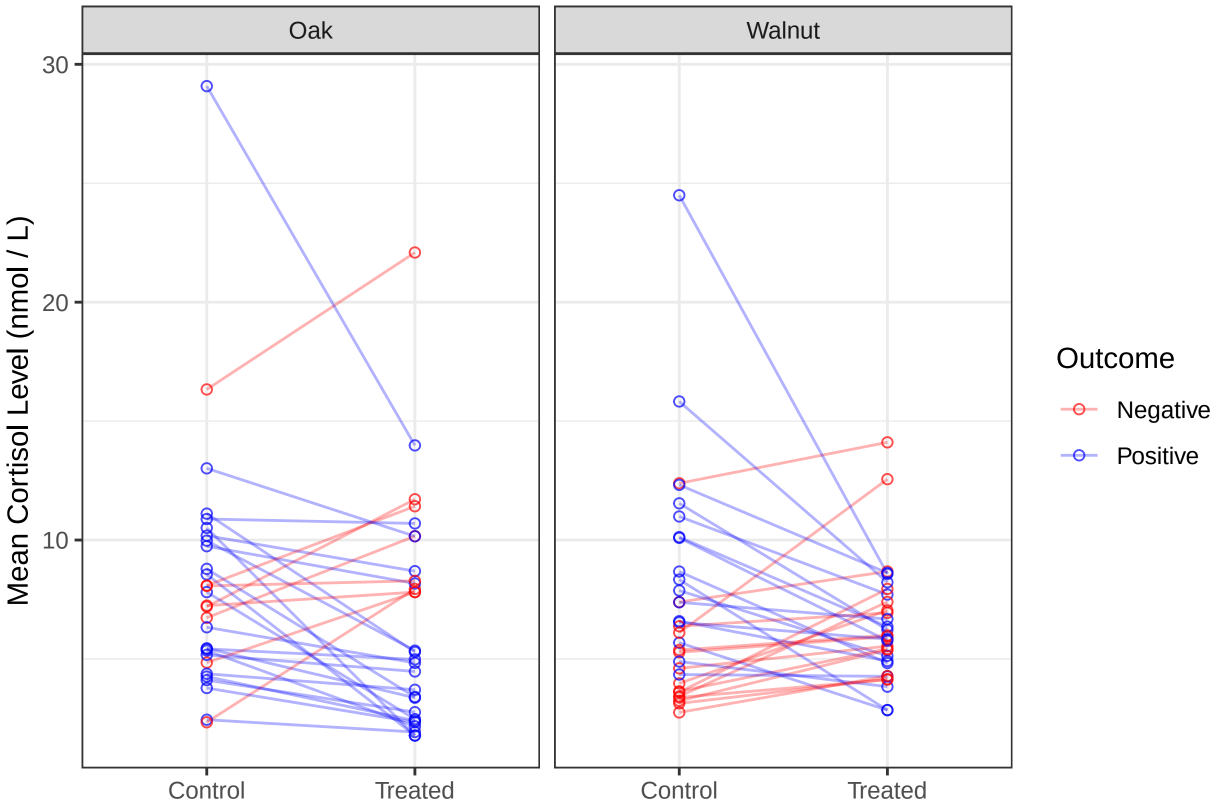
```
r1.re.per <- bind_rows(r1.re.oak, r1.re.wal)
r1.re.per$Response <- r1.re.per$Control - r1.re.per$Treated
r1.re.per$Response.per <- (r1.re.per$Control - r1.re.per$Treated) / r1.re.per$Control
r1.re.per$Outcome <- ifelse(r1.re.per$Response > 0, "Positive", "Negative")
r1.re.per %>% filter(abs(Response.per) > 1) %>% group_by(Room, Outcome) %>%
  summarise(mu=mean(Response.per), sd=sd(Response.per),
          min=min(Response.per), max=max(Response.per), n=n())
```

Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-1.094919	NA	-1.094919	-1.094919	1
Oak	Negative	-1.252845	NA	-1.252845	-1.252845	1

Check for differences within-subjects in initial 3 samples

If these are different this confuses the outcome slightly.

```
#mean per subject for periods intervals 123

r1.re <- r1 %>% group_by(Anon, TestType, Room) %>% filter(Interval %in% c(1,2,3)) %>%
  summarise(Test.mu=mean(Cortisol.fitted.nmol, na.rm=TRUE))

r1.re <- r1.re %>% group_by(Anon) %>% mutate(testDif=Test.mu - lag(Test.mu))
r1.re$Outcome <- ifelse(is.na(r1.re$testDif), "X",
                 ifelse(r1.re$testDif < 0, "Positive",
                  "Negative"))
r1.re$Outcome <- ifelse(r1.re$Outcome == "X", lead(r1.re$Outcome), r1.re$Outcome)
r1.re$Label <- ifelse(r1.re$Room == "A-Walnut", "Walnut", "Oak")
ggplot(data=r1.re, aes(x=TestType, y=Test.mu, colour=Outcome, group=Anon)) +
  theme_bw() +
  geom_line(alpha=.3) +
  geom_point(alpha=.7, shape=1) +
  facet_wrap(~Label) +
  scale_colour_manual(values=c("#FF0000", "#0000FF")) +
  labs(x="", y="Mean Cortisol Level (nmol / L)")
```



```
#create seperate data frames for each room
r1.re.oak <- r1.re %>% filter(Room == "Oak") %>%
  select(Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

r1.re.wal <- r1.re %>% filter(Room == "A-Walnut") %>%
  select(Anon, Room, TestType, Test.mu) %>%
  ungroup() %>% spread(TestType, Test.mu)

#hist(r1.re.wal$Control)
#hist(r1.re.wal$Treated)
#hist(r1.re.oak$Control)
#hist(r1.re.oak$Treated)
#the data aren't normal enough, even with a log transform
#We use Wilcoxon rank sum, we're within-subjects so it's a paired test
#and with our parameterisation our alternative two-sided since this is a post-hoc test.
wilcox.test(r1.re.oak$Control, r1.re.oak$Treated, paired=TRUE,
            alternative="two.sided", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
##
## data:  r1.re.oak$Control and r1.re.oak$Treated
## V = 306, p-value = 0.05623
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.05397879  3.00660763
## sample estimates:
## (pseudo)median
##       1.276776
```

```
wilcox.test(r1.re.wal$Control, r1.re.wal$Treated, paired=TRUE,
            alternative="two.sided", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
```

```
##
## data:  r1.re.wal$Control and r1.re.wal$Treated
## V = 297, p-value = 0.3468
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.6839427  2.0169411
## sample estimates:
## (pseudo)median
##       0.766541
```

```
r1.re %>% filter(!is.na(testDif)) %>% group_by(Room, Outcome) %>%
  mutate(realtd=testDif*-1) %>% summarise(mu=mean(realtd), sd=sd(realtd),
                                          min=min(realtd), max=max(realtd), n=n())
```

Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-2.055427	1.703267	-6.4538212	-0.5646980	15
A-Walnut	Positive	3.940227	3.775952	0.0765959	15.9198309	16
Oak	Negative	-3.298104	2.067915	-5.7558608	-0.2139715	8
Oak	Positive	3.409692	3.585542	0.1848818	15.1039944	21

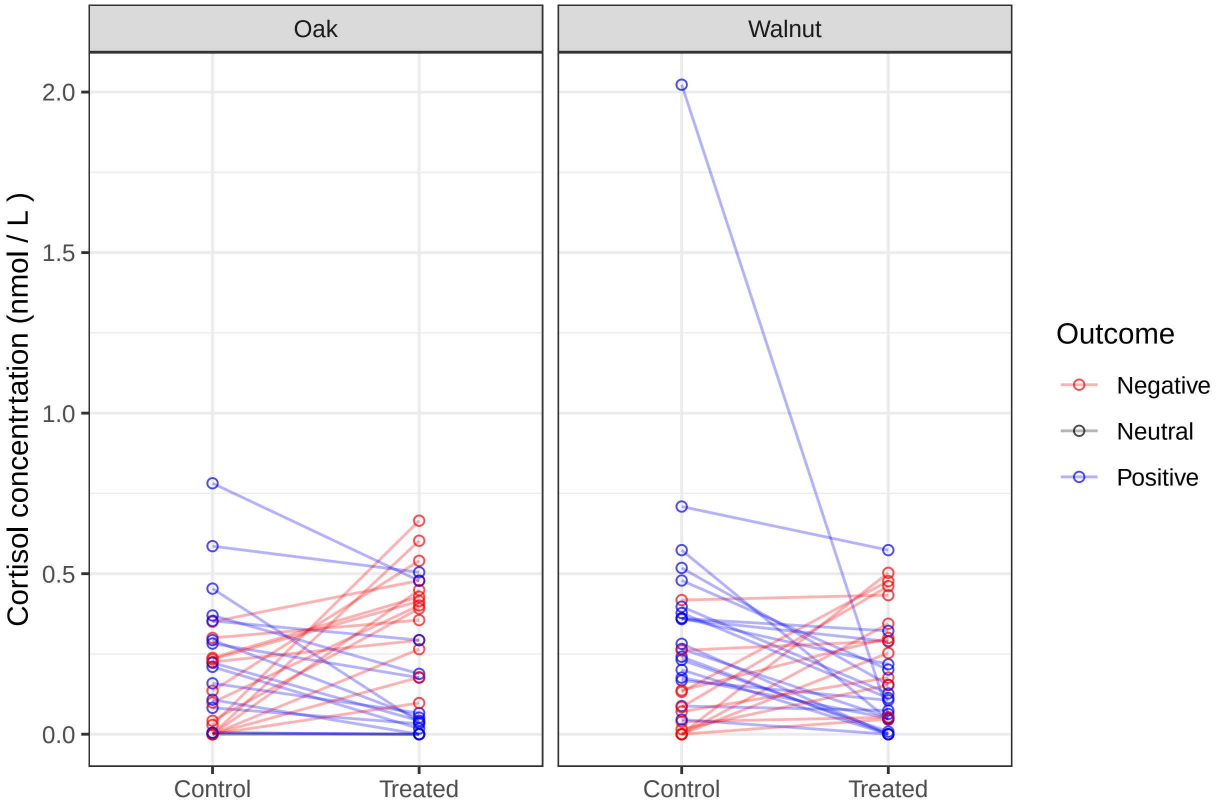
```
r1.re.per <- bind_rows(r1.re.oak, r1.re.wal)
r1.re.per$Response <- r1.re.per$Control - r1.re.per$Treated
r1.re.per$Response.per <- (r1.re.per$Control - r1.re.per$Treated) / r1.re.per$Control
r1.re.per$Outcome <- ifelse(r1.re.per$Response > 0, "Positive", "Negative")
r1.re.per %>% filter(abs(Response.per) > 1) %>% group_by(Room, Outcome) %>%
  summarise(mu=mean(Response.per), sd=sd(Response.per), min=min(Response.per),
           max=max(Response.per), n=n())
```

Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-1.140045	0.0754844	-1.204365	-1.056943	3
Oak	Negative	-2.395035	NA	-2.395035	-2.395035	1

Response magnitude

This is tricky. We find the minimum of either stage 4 or 5, then the maximum after 4 (could be the same as the minimum indicating no stress response detected).

```
r1.min <-  r1 %>% group_by(Anon,TestType, Room) %>% filter(Interval %in% c(4,5)) %>%
  filter(Cortisol.fitted.nmol == min(Cortisol.fitted.nmol)) %>% arrange(Anon) %>%
  mutate(Value="Min")
r1.max <- r1 %>% group_by(Anon,TestType, Room) %>% filter(Interval > 4) %>%
  filter(Cortisol.fitted.nmol == max(Cortisol.fitted.nmol)) %>% arrange(Anon) %>%
  mutate(Value="Max")

r1.min.oak <- r1.min %>% filter(Room == "Oak") %>%
  select(Anon, TestType, Room, Cortisol.fitted.nmol)
names(r1.min.oak)[4] <- "Min.cortisol.nmol"

r1.max.oak <- r1.max %>% filter(Room == "Oak") %>%
  select(Anon, TestType, Room, Cortisol.fitted.nmol)

r1.mm.oak <- r1.min.oak
r1.mm.oak$Max.cortisol.nmol <- r1.max.oak$Cortisol.fitted.nmol
#r1.mm.oak$Interval.max <- r1.max.oak$Interval

r1.mm.oak$Response <- r1.mm.oak$Max.cortisol.nmol - r1.mm.oak$Min.cortisol.nmol
r1.mm.oak$Response.per <- (r1.mm.oak$Max.cortisol.nmol - r1.mm.oak$Min.cortisol.nmol) /
  r1.mm.oak$Min.cortisol.nmol

r1.mm.oak.tst <- r1.mm.oak %>% ungroup() %>% select(Anon, TestType, Response.per, Room) %>%
  spread(TestType, Response.per)

r1.min.wal <- r1.min %>% filter(Room == "A-Walnut") %>%
  select(Anon, TestType, Room, Cortisol.fitted.nmol)
names(r1.min.wal)[4] <- "Min.cortisol.nmol"


r1.max.wal <- r1.max %>% filter(Room == "A-Walnut") %>%
  select(Anon, TestType, Room, Cortisol.fitted.nmol)

r1.mm.wal <- r1.min.wal
r1.mm.wal$Max.cortisol.nmol <- r1.max.wal$Cortisol.fitted.nmol


r1.mm.wal$Response <- r1.mm.wal$Max.cortisol.nmol - r1.mm.wal$Min.cortisol.nmol
r1.mm.wal$Response.per <- (r1.mm.wal$Max.cortisol.nmol - r1.mm.wal$Min.cortisol.nmol) /
  r1.mm.wal$Min.cortisol.nmol

r1.mm.wal.tst <- r1.mm.wal %>% ungroup() %>% select(Anon, TestType, Response.per, Room) %>%
  spread(TestType, Response.per)

#visualisation
r1.mm <- bind_rows(r1.mm.oak.tst, r1.mm.wal.tst)
r1.mm$diff <- r1.mm$Control - r1.mm$Treated
r1.mm$Outcome <- ifelse(r1.mm$diff == 0, "Neutral",
                        ifelse(r1.mm$diff > 0, "Positive", "Negative"))
r1.mm <- r1.mm %>% gather("TestType", "Response", 3:4)
r1.mm$Labels <- ifelse(r1.mm$Room == "A-Walnut", "Walnut", "Oak")


ggplot(data=r1.mm, aes(x=TestType, y=Response, colour=Outcome, group=Anon)) +
  theme_bw() +
  geom_line(alpha=.3) +
  geom_point(alpha=.7, shape=1) +
  facet_wrap(~Labels) +
  scale_y_continuous() +
  scale_colour_manual(values=c("#FF0000", "#000000", "#0000FF")) +
  labs(x="", y="Cortisol concentrtation (nmol / L )")
```



```
#the data aren't normal enough, even with a log transform
#We use Wilcoxon rank sum, we're within-subjects so it's a paired test
#and with our parameterisation our alternative in this case is greater.
wilcox.test(r1.mm.oak.tst$Control, r1.mm.oak.tst$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
## Warning in wilcox.test.default(r1.mm.oak.tst$Control,
## r1.mm.oak.tst$Treated, : cannot compute exact p-value with zeroes

## Warning in wilcox.test.default(r1.mm.oak.tst$Control,
## r1.mm.oak.tst$Treated, : cannot compute exact confidence interval with
## zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  r1.mm.oak.tst$Control and r1.mm.oak.tst$Treated
## V = 161, p-value = 0.8334
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.1492225        Inf
## sample estimates:
## (pseudo)median
##    -0.05446128
```

```
wilcox.test(r1.mm.wal.tst$Control, r1.mm.wal.tst$Treated, paired=TRUE,
            alternative="greater", conf.int=TRUE, conf.level=.95)
```

```
##
##  Wilcoxon signed rank test
##
```

```
## data:  r1.mm.wal.tst$Control and r1.mm.wal.tst$Treated
## V = 315, p-value = 0.0977
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.01963834         Inf
## sample estimates:
## (pseudo)median
##     0.06464738
```

```
#Percent base
r1.mm %>% filter(TestType == "Treated") %>% group_by(Room, Outcome) %>%
  summarise(mu=mean(diff), sd=sd(diff), min=min(diff),
            max=max(diff), n=n())
```

Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-0.1944360	0.1658412	-0.5026537	-0.0135759	12
A-Walnut	Positive	0.2873899	0.4245624	0.0136866	1.9599866	19
Oak	Negative	-0.2790972	0.1882991	-0.6363394	-0.0555621	14
Oak	Neutral	0.0000000	NA	0.0000000	0.0000000	1
Oak	Positive	0.1444287	0.1171052	0.0039665	0.4158734	14

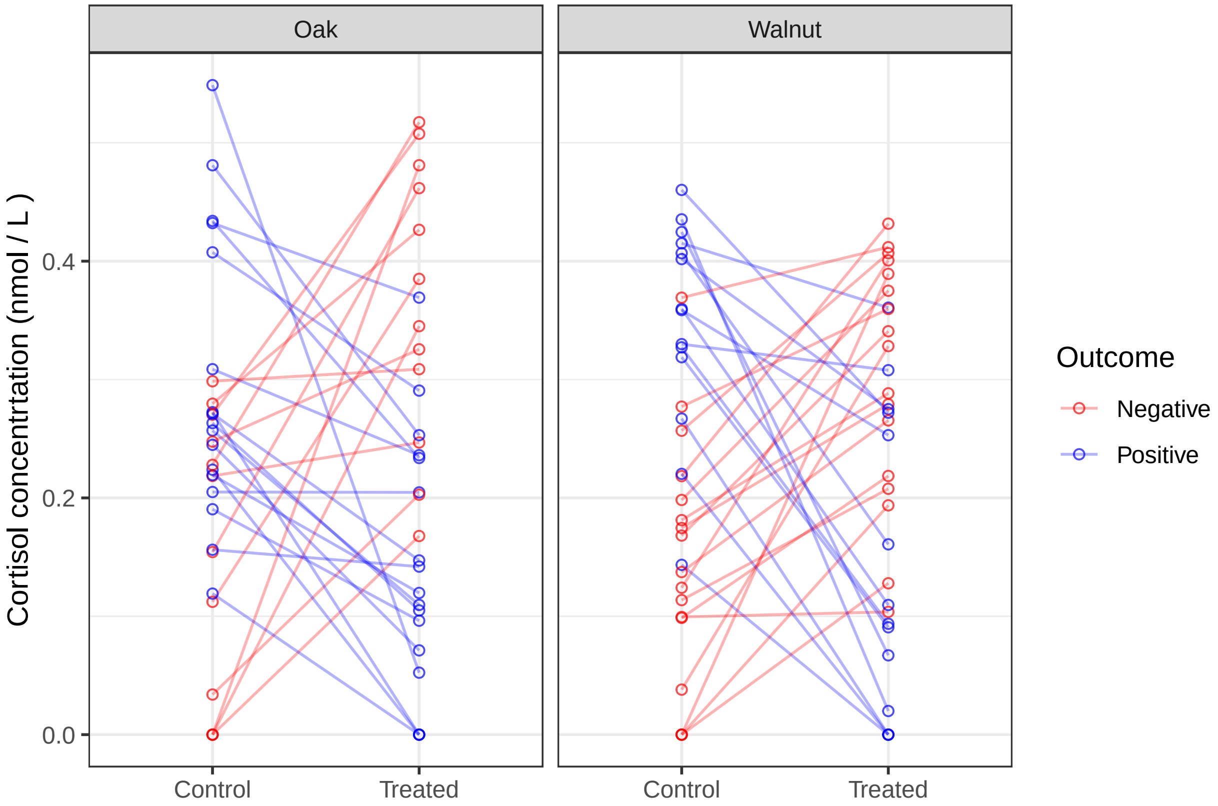
```
r1.mm %>% group_by(Room) %>% filter(TestType == "Treated") %>%
  summarise(mu=mean(diff), sd=sd(diff),
            min=min(diff), max=max(diff), n=n())
```

Room	mu	sd	min	max	n
A-Walnut	0.1008767	0.4185119	-0.5026537	1.9599866	31
Oak	-0.0650124	0.2604398	-0.6363394	0.4158734	29

Recovery

To parameterise recovery, we take the max value of intervals 4, 5, 6, and 7, and the final value (interval 7). If those two are equal (i.e., the max value is at interval 7), that means recovery didn't take place in the test period. We compare as a percent of the max value returned, rather than raw differences.

```r
r1.rec <- r1 %>% group_by(Anon,TestType, Room) %>% filter(Interval > 3) %>%
  filter(Cortisol.fitted.nmol == max(Cortisol.fitted.nmol)) %>%
  arrange(Anon, TestType)

r1.rec7 <- r1 %>% group_by(Anon,TestType, Room) %>% filter(Interval == 7) %>%
  arrange(Anon, TestType)

r1.rec <- r1.rec %>% select(Anon, Interval, TestType, Room, Cortisol.fitted.nmol)
r1.rec$Final <- r1.rec7$Cortisol.fitted.nmol
names(r1.rec)[5] <- "Max"

r1.rec$RawDiff <- r1.rec$Max - r1.rec$Final
r1.rec$PerDiff <- (r1.rec$Max - r1.rec$Final) / r1.rec$Max

r1.rec.per.o <- r1.rec %>% filter(Room == "Oak") %>%
  select(Anon, TestType, PerDiff, Room) %>%
  spread(TestType, PerDiff)
r1.rec.per.w <- r1.rec %>% filter(Room == "A-Walnut") %>%
  select(Anon, TestType, PerDiff, Room) %>%
  spread(TestType, PerDiff)

r1.rec.raw.o <- r1.rec %>% filter(Room == "Oak") %>%
  select(Anon, TestType, RawDiff, Room) %>%
  spread(TestType, RawDiff)
r1.rec.raw.w <- r1.rec %>% filter(Room == "A-Walnut") %>%
  select(Anon, TestType, RawDiff, Room) %>%
  spread(TestType, RawDiff)

#visualisation
r1.recv <- bind_rows(r1.rec.per.o, r1.rec.per.w)
r1.recv$diff <- r1.recv$Control - r1.recv$Treated
r1.recv$Outcome <- ifelse(r1.recv$diff == 0, "Neutral",
                        ifelse(r1.recv$diff > 0, "Positive", "Negative"))
r1.recv <- r1.recv %>% gather("TestType", "Response", 3:4)
r1.recv$Labels <- ifelse(r1.recv$Room == "A-Walnut", "Walnut", "Oak")

ggplot(data=r1.recv, aes(x=TestType, y=Response, colour=Outcome, group=Anon)) +
  theme_bw() +
  geom_line(alpha=.3) +
  geom_point(alpha=.7, shape=1) +
  facet_wrap(~Labels) +
  scale_y_continuous() +
  scale_colour_manual(values=c("#FF0000", "#0000FF")) +
  labs(x="", y="Cortisol concentrtation (nmol / L )")
```



```
#parameterisation changes here:
#we're looking at recovery, and our hypothesis is that one recovers less
#in the control room. We keep greater, and reverse the order we specify control and
#treated
wilcox.test(r1.rec.per.o$Treated, r1.rec.per.o$Control, paired=TRUE,
            alternative="greater", conf.int = TRUE, conf.level = .95, correct=FALSE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  r1.rec.per.o$Treated and r1.rec.per.o$Control
## V = 211, p-value = 0.5592
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.09106038         Inf
## sample estimates:
## (pseudo)median
##   -0.007283567
```

```
wilcox.test(r1.rec.per.w$Treated, r1.rec.per.w$Control, paired=TRUE,
            alternative="greater", conf.int = TRUE, conf.level = .95, correct=FALSE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  r1.rec.per.w$Treated and r1.rec.per.w$Control
## V = 238, p-value = 0.5804
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -0.06888677         Inf
## sample estimates:
## (pseudo)median
##   -0.008325758
```

```
r1.rec.per <- bind_rows(r1.rec.per.w, r1.rec.per.o)
r1.rec.per$Resp.diff <- r1.rec.per$Treated - r1.rec.per$Control
```

```
r1.rec.per$Outcome <- ifelse(r1.rec.per$Resp.diff > 0, "Positive", "Negative")

r1.rec.per %>%  group_by(Room, Outcome) %>%
  summarise(mu=mean(Resp.diff), sd=sd(Resp.diff), min=min(Resp.diff),
            max=max(Resp.diff), n=n())
```

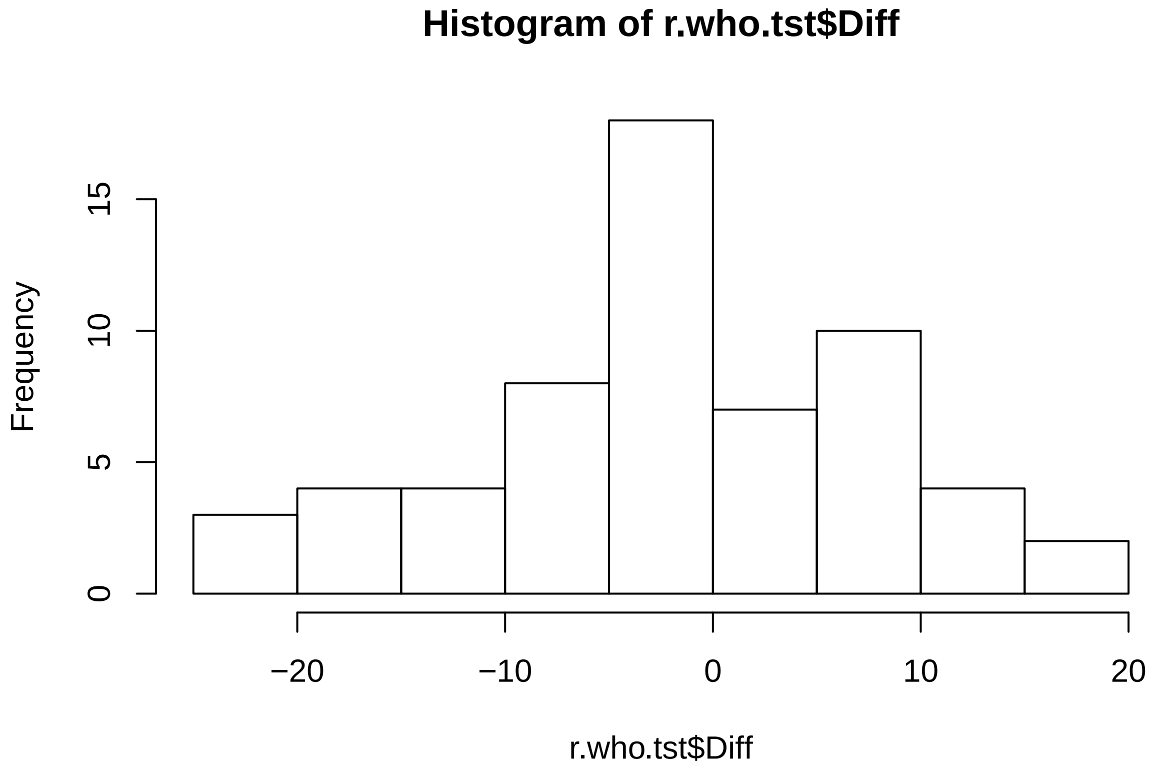
Room	Outcome	mu	sd	min	max	n
A-Walnut	Negative	-0.2041792	0.1084512	-0.4153457	-0.0220249	14
A-Walnut	Positive	0.1572617	0.0953268	0.0043170	0.3893027	17
Oak	Negative	-0.1531754	0.1151816	-0.4963546	-0.0003141	17
Oak	Positive	0.2110977	0.1381763	0.0101800	0.4810324	12

```
r1.rec.per %>%  group_by(Room) %>%
  summarise(mu=mean(Resp.diff), sd=sd(Resp.diff), min=min(Resp.diff),
            max=max(Resp.diff), n=n())
```

Room	mu	sd	min	max	n
A-Walnut	-0.0059697	0.2082683	-0.4153457	0.3893027	31
Oak	-0.0024417	0.2200429	-0.4963546	0.4810324	29

#WHO-5 Well-being index

```
r.who.tst <- r1 %>% select(Anon, Test, TestType, Room, WHO5_Total) %>%
  group_by(Anon, TestType, Room) %>% summarise(WHO5_Total = mean(WHO5_Total*4)) %>%
  spread(TestType, WHO5_Total) %>% mutate(Diff = (Treated) - (Control))

r.who.tst$Outcome <- ifelse(r.who.tst$Diff == 0, "Neutral",
                 ifelse(r.who.tst$Diff > 0, "Positive", "Negative"))

hist(r.who.tst$Diff)
```



```
t.test(r.who.tst$Control, r.who.tst$Treated, paired=TRUE, conf.level = 0.95)
```

```
##
```

```
##  Paired t-test
##
## data:  r.who.tst$Control and r.who.tst$Treated
## t = 1.0551, df = 59, p-value = 0.2957
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.195363  3.862030
## sample estimates:
## mean of the differences
##                1.333333
```

```
r.who <- r1 %>% select(Anon, Test, TestType, Room, WHO5_Total) %>%
  mutate(WHO5_100 = WHO5_Total *4)

r.who %>% group_by(Room, TestType) %>%
  summarise(min=min(WHO5_100), max=max(WHO5_100), med=median(WHO5_100),
            mean=mean(WHO5_100), sd=sd(WHO5_100))
```

Room	TestType	min	max	med	mean	sd
A-Walnut	Control	40	100	68	68.90323	12.41843
A-Walnut	Treated	32	100	68	65.67742	15.47002
Oak	Control	24	92	64	62.06897	18.40940
Oak	Treated	28	88	68	62.75862	16.24917

```
r.who.tst %>% group_by(Room) %>%
  summarise(min=min(Diff), max=max(Diff), med=median(Diff),
            mean=mean(Diff), sd=sd(Diff))
```

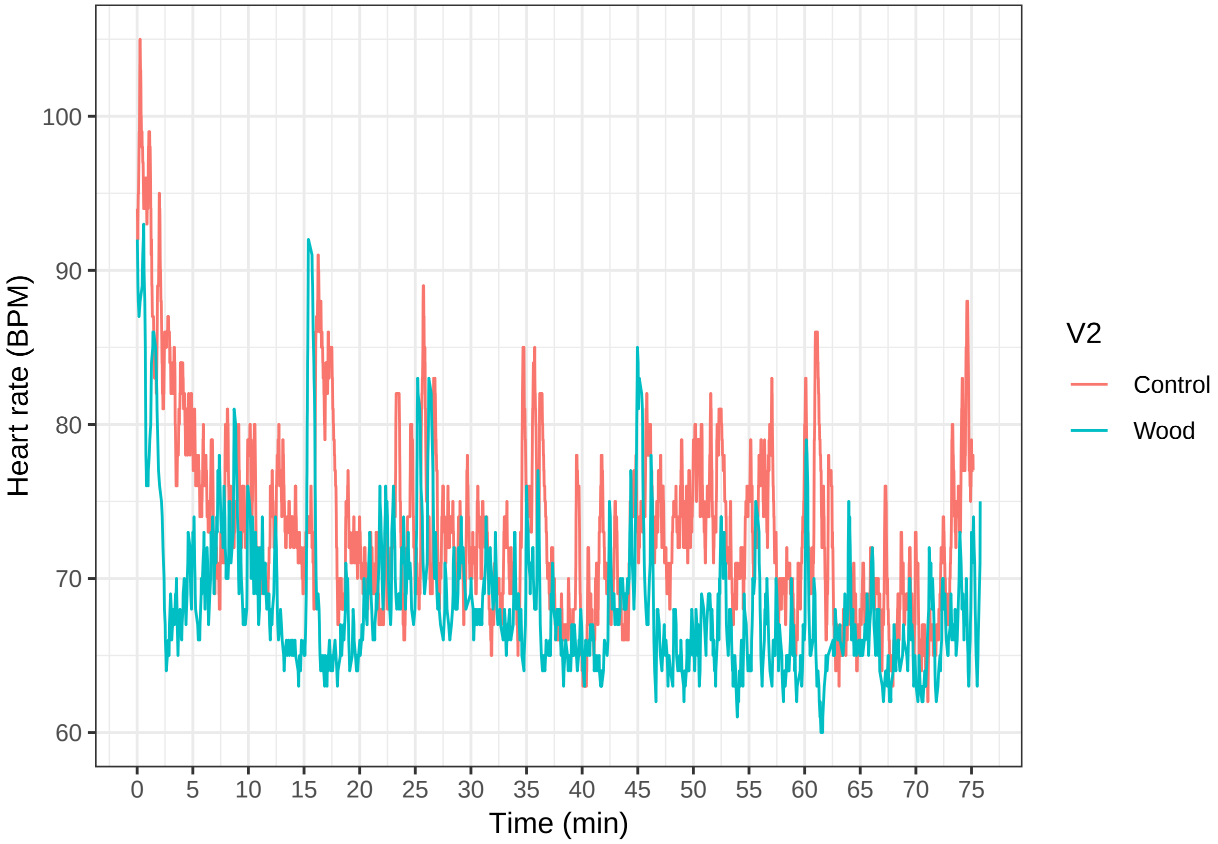
Room	min	max	med	mean	sd
A-Walnut	-24	12	0	-3.2258065	10.193821
Oak	-16	20	0	0.6896552	9.075491

Heart rate

One example.

```
px.c <- read.csv("data/garminxml/garmindats_ANON-Control.txt", header = FALSE,
                 stringsAsFactors=FALSE)
px.w <- read.csv("data/garminxml/garmindats_ANON-Wood.txt", header = FALSE,
                 stringsAsFactors=FALSE)

px.c$datetime <- as.POSIXct(px.c$V4)
px.w$datetime <- as.POSIXct(px.w$V4)
px.c <- px.c[complete.cases(px.c),]
px.w <- px.w[complete.cases(px.w),]
px.c <- px.c %>% select(V1, V2, V7, datetime) %>%
  mutate(timedif = as.numeric(difftime(datetime, lag(datetime, default=0))))
px.c[1,5] <- 1
px.c <- px.c %>% mutate(elapsed = cumsum(timedif), elapsed.min = cumsum(timedif/60))

px.w <- px.w %>% select(V1, V2, V7, datetime) %>%
  mutate(timedif = as.numeric(difftime(datetime, lag(datetime, default=0))))

px.w[1,5] <- 1
px.w <- px.w %>% mutate(elapsed = cumsum(timedif), elapsed.min = cumsum(timedif/60))

px <- bind_rows(px.c, px.w)
```

```
ggplot(data=px, aes(x=elapsed.min, y=V7, colour=V2)) + theme_bw() +
  geom_line() +
  scale_x_continuous(breaks=seq(0,75,5)) +
  labs(x="Time (min)",
       y=("Heart rate (BPM)"))
```



#Environment

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2  zoo_1.8-6       scales_1.0.0    ggforce_0.2.2
##  [5] forcats_0.4.0   stringr_1.4.0   dplyr_0.8.1     purrr_0.3.2
##  [9] readr_1.3.1     tidyr_0.8.3     tibble_2.1.2    ggplot2_3.1.1
## [13] tidyverse_1.2.1 drc_3.0-1       MASS_7.3-51.4
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.0        jsonlite_1.6      splines_3.6.0
##  [4] carData_3.0-2     modelr_0.1.4      gtools_3.8.1
##  [7] assertthat_0.2.1  highr_0.8         cellranger_1.1.0
## [10] yaml_2.2.0        pillar_1.4.1      backports_1.1.4
## [13] lattice_0.20-38   glue_1.3.1        digest_0.6.19
## [16] polyclip_1.10-0   rvest_0.3.4       colorspace_1.4-1
## [19] sandwich_2.5-1    htmltools_0.3.6   Matrix_1.2-17
## [22] plyr_1.8.4        pkgconfig_2.0.2   broom_0.5.2
## [25] haven_2.1.0       mvtnorm_1.0-10    tweenr_1.0.1
## [28] openxlsx_4.1.0.1  rio_0.5.16        generics_0.0.2
## [31] farver_1.1.0      car_3.0-3         TH.data_1.0-10
## [34] withr_2.1.2       lazyeval_0.2.2    cli_1.1.0
## [37] survival_2.44-1.1 magrittr_1.5      crayon_1.3.4
## [40] readxl_1.3.1      evaluate_0.14     nlme_3.1-140
## [43] xml2_1.2.0        foreign_0.8-71    tools_3.6.0
## [46] data.table_1.12.2 hms_0.4.2         multcomp_1.4-10
## [49] munsell_0.5.0     plotrix_3.7-5     zip_2.0.2
## [52] compiler_3.6.0    rlang_0.3.4       grid_3.6.0
## [55] rstudioapi_0.10   labeling_0.3      rmarkdown_1.13
## [58] gtable_0.3.0      codetools_0.2-16  abind_1.4-5
## [61] curl_3.3          R6_2.4.0          lubridate_1.7.4
## [64] knitr_1.23        bindr_0.1.1       stringi_1.4.3
## [67] Rcpp_1.0.1        tidyselect_0.2.5  xfun_0.7
```