R Notebook

This R Markdown (http://rmarkdown.rstudio.com) Notebook runs code for the tables and figures presented in the paper: “Wired to exit: Exploring the effects of wayfinding affordances in underground facilities using Virtual Reality”

```
library(plyr)
library(readr)
library(dplyr)
library(tidyverse)
```

```
#Import and merge individual HR data files from the ./raw folder
files <- list.files("raw",full.names=TRUE)
tbl <- sapply(files, read_csv,col_name =c("time", "value"), simplify=FALSE) %>%
  bind_rows(.id = "id")

tbl$id<-stringr::str_replace(tbl$id, "raw/", "")
```

```
dim(tbl)
```

```
## [1] 21083     3
```

```
names(tbl)
```

```
## [1] "id"    "time"  "value"
```

```
table(tbl$id)
```

```
##
## user10 user11 user12 user13 user14 user15 user16 user17 user18 user19
##   1225   1527   1237   1262   1101   1348   1261   1493   1617   1002
## user20  user4  user5  user6  user7  user8  user9
##   1770   1020    610    994   1238   1259   1119
```

```
tbl
```

id <chr>	time <S3: POSIXct>	value <dbl>
user10	2020-02-11 10:16:48	70
user10	2020-02-11 10:16:49	68
user10	2020-02-11 10:16:50	68
user10	2020-02-11 10:16:51	68
user10	2020-02-11 10:16:52	68
user10	2020-02-11 10:16:53	68
user10	2020-02-11 10:16:54	68
user10	2020-02-11 10:16:55	69
user10	2020-02-11 10:16:56	69
user10	2020-02-11 10:16:57	70

1-10 of 10,000 rows   Previous **1** 2 3 4 5 6 ... 1000 Next

```
#Convert to correct time zone
attr(tbl$time, "tzone") <- "Europe/Helsinki"

#Create new feature for relative time
tbl<- tbl %>% group_by(id) %>% mutate(counter = row_number())

#Assign individual levels
tbl$id = factor(tbl$id, levels = c('user4', 'user5', "user6", "user7",
                                   "user8", "user9",'user10', 'user11', "user12", "user13",
                                   "user14", "user15","user16", "user17","user18","user19", "user20"),
                labels = c(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20))
```

```
dim(tbl)
```

```
## [1] 21083     4
```

```
tbl
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
10	2020-02-11 12:16:48	70	1
10	2020-02-11 12:16:49	68	2
10	2020-02-11 12:16:50	68	3
10	2020-02-11 12:16:51	68	4
10	2020-02-11 12:16:52	68	5
10	2020-02-11 12:16:53	68	6
10	2020-02-11 12:16:54	68	7
10	2020-02-11 12:16:55	69	8
10	2020-02-11 12:16:56	69	9
10	2020-02-11 12:16:57	70	10

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
#Import Physiological & Behavioral indicators
data <- read.delim("~/Desktop/data_BMP/data.tsv")

#Convert date-time
data$StartTime<- strptime(data$StartTime, format="%m.%d.%Y %H.%M.%S")
data$EndTime<- strptime(data$EndTime, format="%m.%d.%Y %H.%M.%S")
```

```
options(dplyr.width =110)
glimpse(data)
```

```
## Observations: 17
## Variables: 106
## $ User.                                                          <int> 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,…
## $ StartTime                                                      <dttm> 2020-02-10 14:52:56, 2020-02-10 15:1…
## $ EndTime                                                        <dttm> 0020-02-10 14:58:58, 0020-02-10 15:1…
## $ Group                                                          <fct> B, A, B, B, B, A, B, B, A, A, B, A, A…
## $ Scene.type                                                     <fct> B, A, C, B, B, A, C, C, A, A, B, A, A…
## $ bmi2                                                           <fct> "20,20", "19,30", "28,70", "26,20", "…
## $ BMI                                                            <fct> "20,2", "19,3", "28,7", "26,2", "20,9…
## $ Age                                                            <int> 28, 23, 27, 24, 26, 28, 21, 25, 22, 2…
## $ Height..cm.                                                    <fct> "1,65", "1,61", "1,72", "1,77", "1,65…
## $ Weight..kg.                                                    <int> 55, 50, 85, 82, 57, 75, 70, 65, 85, 4…
## $ Gender                                                         <fct> Female, Female, Male, Male, Female, M…
## $ Student.University.Employee.Visitor                            <fct> Student, Student, Student, Student, S…
## $ Major.Profession                                               <fct> Education, Student PhD, Electrical En…
## $ Military.Service.division.                                     <fct> No, No, Yes, No, No, No, No, No, No, …
## $ Voluntary.rescue.firefighting.                                 <fct> No, No, No, No, No, No, No, No, No, N…
## $ Do.you.have.a..driver.s.license.                               <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Ye…
## $ Do.you.drive.a.car.                                            <fct> No, No, No, Yes, No, No, No, No, No, …
## $ How.often.do.you.park.to.an.underground.parking.space.         <fct> seldom, seldom, seldom, often, seldom…
## $ How.often.do.you.play.video.games.                             <fct> Never, Never, Daily, Never, Never, Da…
## $ How.often.do.you.use.virtual.reality.equipment.                <fct> Seldom/Never, Seldom/Never, Seldom/Ne…
## $ X                                                              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ I.am.very.good.at.giving.directions                            <int> 2, 2, 3, 2, 1, 3, 2, 3, 3, 1, 2, 3, 2…
## $ I.think.it.is.important.to.find.new.routes.in.the.environment <int> 2, 2, 2, 3, 3, 4, 3, 3, 3, 3, 3, 3, 2…
## $ I.have.a.poor.memory.for.where.I.left.things                   <int> 3, 1, 1, 3, 2, 1, 0, 2, 0, 2, 2, 2, 3…
## $ I.like.to.travel                                               <int> 4, 3, 2, 4, 3, 3, 4, 3, 4, 3, 4, 4, 2…
## $ I.am.very.good.at.judging.distances                            <int> 2, 2, 3, 3, 3, 3, 1, 1, 2, 1, 3, 3, 3…
## $ My..sense.of.direction..is.very.poor                           <int> 3, 3, 0, 1, 2, 2, 0, 2, 1, 3, 1, 0, 1…
## $ X.1                                                            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ Game.statistics                                                <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ Needed.assistance.                                             <fct> Yes, No, No, No, No, No, No, No, No, …
## $ Completion.time..s.                                            <int> 362, 141, 168, 82, 112, 92, 101, 243,…
## $ Walk.distance..m.                                              <int> 900, 306, 556, 241, 343, 426, 341, 65…
## $ Avg.speed..m.s.                                                <fct> "2,48", "2,17", "3,32", "2,91", "3,07…
## $ Number.of.stops                                                <int> 72, 35, 31, 13, 37, 7, 4, 16, 22, 21,…
## $ Number.of.rotations..45.deg.turns.                             <int> 553, 704, 109, 60, 100, 301, 62, 138,…
## $ Rotations.in.degrees                                           <int> 24885, 31680, 4905, 2700, 4500, 13545…
## $ Ambulance.found.                                               <fct> YES, YES, YES, YES, YES, YES, YES, YE…
```

```
## $ Ambulance.time                                     <int> 36, 17, 33, 15, 54, 10, 21, 19, 17, 3…
## $ X.2                                                <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ While.playing                                      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ I.was.interested.in.the.game.s.story               <int> 2, 2, 1, 3, 4, 3, 2, 1, 2, 3, 3, 3, 2…
## $ I.felt.succesful                                   <int> 2, 3, 3, 4, 4, 4, 4, 2, 4, 2, 3, 1, 3…
## $ I.felt.bored                                       <int> 1, 0, 1, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0…
## $ I.found.it.impressive                              <int> 3, 3, 1, 2, 2, 3, 3, 2, 3, 3, 2, 2, 3…
## $ I.forgot.everything.around.me                      <fct> 3, 0, 2, n/a, 2, 1, 0, 2, 3, 1, 1, 1,…
## $ I.felt.frustrated                                  <int> 3, 0, 0, 1, 2, 0, 0, 2, 1, 0, 1, 0, 1…
## $ I.found.it.tiresome                                <int> 3, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1…
## $ I.felt.irritable                                   <int> 3, 0, 1, 2, 3, 0, 0, 1, 1, 0, 2, 0, 0…
## $ I.felt.skillful                                    <fct> 2, 1, 2, n/a, 2, 4, 1, 1, 3, 2, 2, 2,…
## $ I.felt.completely.absorbed                         <int> 3, 2, 2, 1, 2, 3, 1, 2, 2, 2, 2, 3, 1…
## $ I.felt.content                                     <int> 2, 2, 2, 1, 2, 3, 3, 1, 3, 2, 3, 2, 2…
## $ I.felt.challnged                                   <int> 3, 3, 3, 0, 4, 1, 1, 2, 3, 2, 3, 3, 2…
## $ I.had.to.put.a.lot.of.effort.into.it               <int> 3, 2, 3, 0, 4, 1, 0, 2, 3, 1, 1, 3, 2…
## $ I.felt.good                                        <int> 2, 2, 2, 3, 1, 4, 4, 2, 4, 3, 3, 4, 3…
## $ I.felt.content.1                                   <int> 2, 2, 2, 1, 2, 3, 3, 2, 3, 2, 2, 3, 2…
## $ I.felt.skillful.1                                  <int> 2, 2, 3, 4, 2, 4, 2, 1, 3, 2, 2, 2, 1…
## $ I.was.interested.in.the.game.s.story.1             <int> 2, 1, 0, 3, 4, 3, 2, 1, 2, 3, 3, 3, 2…
## $ I.thought.it.was.fun                               <int> 2, 3, 1, 2, 0, 3, 4, 3, 4, 3, 3, 1, 2…
## $ I.was.fully.occupied.with.the.game                 <int> 3, 2, 3, 2, 4, 3, 1, 3, 4, 3, 3, 3, 2…
## $ I.felt.happy                                       <int> 2, 1, 1, 2, 0, 3, 4, 1, 3, 3, 3, 3, 2…
## $ It.gave.me.a.bad.mood                              <int> 3, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0…
## $ I.thought.about.other.things                       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1…
## $ I.found.it.tiresome.1                              <int> 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1…
## $ I.felt.competent                                   <int> 1, 2, 2, 3, 3, 4, 2, 2, 3, 2, 3, 2, 2…
## $ I.thought.it.was.hard                              <int> 3, 1, 1, 0, 4, 0, 0, 1, 1, 1, 2, 1, 1…
## $ It.was.aesthetically.pleaseingn                    <int> 1, 3, 0, 3, 2, 2, 3, 2, 3, 3, 3, 2, 2…
## $ I.forgot.about.everything.around.me                <int> 3, 0, 1, 3, 2, 3, 0, 3, 2, 1, 3, 0, 0…
## $ I.felt.good.1                                      <int> 1, 1, 2, 2, 0, 3, 3, 1, 3, 3, 3, 3, 2…
## $ I.was.good.at.it                                   <int> 2, 1, 2, 4, 3, 4, 2, 1, 3, 2, 3, 1, 2…
## $ I.felt.bored.1                                     <int> 1, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0…
## $ I.felt.succesful.1                                 <int> 2, 2, 3, 4, 3, 4, 3, 2, 4, 2, 2, 2, 2…
## $ I.felt.imaginative                                 <int> 2, 2, 1, 3, 2, 3, 2, 0, 2, 2, 3, 3, 1…
## $ I.felt.that.I.could.explore.things                 <int> 2, 2, 3, 3, 2, 4, 4, 1, 4, 3, 4, 3, 3…
## $ I.enjoyed.it                                       <int> 2, 3, 1, 3, 0, 3, 3, 1, 4, 3, 3, 4, 3…
## $ I.was.fast.at.reaching.the.game.s.targets          <int> 2, 3, 3, 4, 4, 4, 2, 1, 4, 3, 3, 2, 1…
## $ I.felt.annoyed                                     <int> 3, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0…
## $ I.felt.pressured                                   <int> 3, 2, 1, 2, 4, 0, 0, 2, 0, 2, 1, 3, 1…
```

```
## $ I.felt.irritable.1                                  <int> 3, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0…
## $ I.lost.track.of.time                                <int> 3, 0, 0, 3, 2, 0, 0, 1, 1, 2, 3, 1, 1…
## $ I.felt.challenged                                   <int> 3, 3, 3, 2, 4, 1, 1, 1, 4, 2, 2, 3, 2…
## $ I.found.it.impressive.1                             <int> 3, 2, 0, 2, 2, 2, 3, 2, 3, 3, 3, 3, 2…
## $ I.was.deeply.concenrated.in.the.game                <int> 3, 3, 1, 2, 4, 3, 2, 3, 4, 2, 3, 4, 3…
## $ I.felt.frustrated.1                                 <int> 3, 0, 1, 0, 2, 0, 0, 2, 0, 0, 1, 0, 1…
## $ If.felt.like.a.rich.experience                      <int> 3, 2, 1, 3, 2, 3, 2, 2, 3, 3, 3, 3, 2…
## $ I.lost.connection.with.the.outside.world            <int> 3, 2, 3, 4, 2, 3, 0, 3, 2, 1, 2, 2, 0…
## $ I.felt.time.pressure                                <int> 3, 2, 1, 2, 3, 0, 0, 2, 0, 3, 3, 4, 3…
## $ I.had.to.put.a.lot.of.effort.into.it.1              <int> 3, 2, 3, 0, 4, 1, 1, 2, 2, 2, 2, 3, 3…
## $ X.3                                                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ After.the.game                                      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ I.felt.revived                                      <int> 1, 1, 1, 2, 4, 3, 1, 1, 2, 2, 1, 2, 3…
## $ I.felt.bad                                          <int> 2, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0…
## $ I.found.it.hard.to.get.back.to.reality              <int> 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ I.felt.guilty                                       <int> 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0…
## $ It.felt.like.a.victory                              <int> 2, 3, 3, 4, 3, 4, 2, 2, 4, 3, 3, 1, 3…
## $ I.found.it.a.waste.of.time                          <int> 2, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0…
## $ I.felt.energized                                    <int> 2, 3, 1, 2, 4, 4, 1, 1, 4, 2, 1, 2, 2…
## $ I.felt.satisfied                                    <int> 2, 3, 1, 2, 4, 4, 3, 2, 4, 3, 2, 3, 3…
## $ I.felt.disoriented                                  <int> 3, 3, 2, 1, 2, 0, 0, 1, 0, 1, 1, 1, 0…
## $ I.felt.exhausted                                    <int> 3, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0…
## $ I.felt.that.I.could.have.done.more.useful.things    <int> 2, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 3, 1…
## $ I.felt.powerful                                     <int> 2, 2, 1, 2, 2, 2, 0, 0, 4, 1, 2, 2, 1…
## $ I.felt.weary                                        <int> 3, 0, 1, 2, 0, 0, 0, 1, 0, 0, 0, 1, 1…
## $ I.felt.regret                                       <int> 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0…
## $ I.felt.ashamed                                      <int> 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0…
## $ I.felt.proud                                        <fct> 2, 3, 1, 3, 2, 4, 2, 1, 4, 2, 2, 1, 2…
## $ I.had.a.sense.that.I.had.returned.form.a.journey    <int> 3, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1…
```

```
# Find the intercept to keep HR data only during gameplay
user4<- tbl %>% filter(id == "4")

data[ ,1:3] # lookup table
```

User.	StartTime	EndTime
<int>	<S3: POSIXlt>	<S3: POSIXlt>

User. <int>	StartTime <S3: POSIXlt>	EndTime <S3: POSIXlt>
4	<POSIXlt>	<POSIXlt>
5	<POSIXlt>	<POSIXlt>
6	<POSIXlt>	<POSIXlt>
7	<POSIXlt>	<POSIXlt>
8	<POSIXlt>	<POSIXlt>
9	<POSIXlt>	<POSIXlt>
10	<POSIXlt>	<POSIXlt>
11	<POSIXlt>	<POSIXlt>
12	<POSIXlt>	<POSIXlt>
13	<POSIXlt>	<POSIXlt>

1-10 of 17 rows   Previous **1** 2 Next

```
user4 %>% filter(time == "2020-2-10 14:52:56") #start time
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
4	2020-02-10 14:52:56	75	626

1 row

```
user4 %>% filter(time == "2020-2-10 14:58:58") #end time
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
4	2020-02-10 14:58:58	86	988

1 row

```
user5<- tbl %>% filter(id == "5")
user5 %>% filter(time == "2020-2-10 15:16:45")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
5	2020-02-10 15:16:45	76	448

1 row

```
user5 %>% filter(time == "2020-2-10 15:19:06")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
5	2020-02-10 15:19:06	69	589

1 row

```
user6<- tbl %>% filter(id == "6")
user6 %>% filter(time == "2020-2-10 15:37:34")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
6	2020-02-10 15:37:34	84	455

1 row

```
user6 %>% filter(time == "2020-2-10 15:40:22")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
6	2020-02-10 15:40:22	87	623

1 row

```
user7<- tbl %>% filter(id == "7")
user7 %>% filter(time == "2020-2-10 15:57:48")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
7	2020-02-10 15:57:48	67	448

1 row

```
user7 %>% filter(time == "2020-2-10 15:59:10")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
7	2020-02-10 15:59:10	65	530

1 row

```
user8<- tbl %>% filter(id == "8")
user8 %>% filter(time == "2020-2-10 16:23:57")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
8	2020-02-10 16:23:57	81	552

1 row

```
user8 %>% filter(time == "2020-2-10 16:25:49")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
8	2020-02-10 16:25:49	80	664

1 row

```
user9<- tbl %>% filter(id == "9")
user9 %>% filter(time == "2020-2-10 16:47:32")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
9	2020-02-10 16:47:32	83	598

1 row

```
user9 %>% filter(time == "2020-2-10 16:49:04")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
9	2020-02-10 16:49:04	105	690

1 row

```
user10<- tbl %>% filter(id == "10")
user10 %>% filter(time == "2020-2-11 12:26:54")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
10	2020-02-11 12:26:54	69	607

1 row

```
user10 %>% filter(time == "2020-2-11 12:28:35")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
10	2020-02-11 12:28:35	72	708

1 row

```
user11<- tbl %>% filter(id == "11")
user11 %>% filter(time == "2020-2-11 12:52:13")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
11	2020-02-11 12:52:13	76	532

1 row

```
user11 %>% filter(time == "2020-2-11 12:56:16")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
11	2020-02-11 12:56:16	72	775

1 row

```
user12<- tbl %>% filter(id == "12")
user12 %>% filter(time == "2020-2-11 13:20:44")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
12	2020-02-11 13:20:44	73	574

1 row

```
user12 %>% filter(time == "2020-2-11 13:22:35")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
12	2020-02-11 13:22:35	80	685

1 row

```
user13<- tbl %>% filter(id == "13")
user13 %>% filter(time == "2020-2-11 13:43:46")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
13	2020-02-11 13:43:46	75	556

1 row

```
user13 %>% filter(time == "2020-2-11 13:46:49")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
13	2020-02-11 13:46:49	73	739

1 row

```
user14<- tbl %>% filter(id == "14")
user14 %>% filter(time == "2020-2-11 14:07:32")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
14	2020-02-11 14:07:32	88	528

1 row

```
user14 %>% filter(time == "2020-2-11 14:09:20")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
14	2020-02-11 14:09:20	89	636

1 row

```
user15<- tbl %>% filter(id == "15")
user15 %>% filter(time == "2020-2-11 14:47:32")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
15	2020-02-11 14:47:32	68	660

1 row

```
user15 %>% filter(time == "2020-2-11 14:48:45")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
15	2020-02-11 14:48:45	62	733

1 row

```
user16<- tbl %>% filter(id == "16")
user16 %>% filter(time == "2020-2-11 15:11:40")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
16	2020-02-11 15:11:40	78	607

1 row

```
user16 %>% filter(time == "2020-2-11 15:14:41")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
16	2020-02-11 15:14:41	94	788

1 row

```
user17<- tbl %>% filter(id == "17")
user17 %>% filter(time == "2020-2-11 15:40:15")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
17	2020-02-11 15:40:15	83	626

1 row

```
user17 %>% filter(time == "2020-2-11 15:44:43")
```

id	time	value	counter
<fctr>	<S3: POSIXct>	<dbl>	<int>
17	2020-02-11 15:44:43	96	894

1 row

```
user18<- tbl %>% filter(id == "18")
user18 %>% filter(time == "2020-2-12 14:38:18")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
18	2020-02-12 14:38:18	85	569

1 row

```
user18 %>% filter(time == "2020-2-12 14:44:47")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
18	2020-02-12 14:44:47	89	958

1 row

```
user19<- tbl %>% filter(id == "19")
user19 %>% filter(time == "2020-2-12 15:05:28")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
19	2020-02-12 15:05:28	96	415

1 row

```
user19 %>% filter(time == "2020-2-12 15:06:28")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
19	2020-02-12 15:06:28	107	475

1 row

```
user20<- tbl %>% filter(id == "20")
user20 %>% filter(time == "2020-2-12 15:42:09")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
20	2020-02-12 15:42:09	120	480

1 row

```
user20 %>% filter(time == "2020-2-12 15:51:27")
```

id <fctr>	time <S3: POSIXct>	value <dbl>	counter <int>
20	2020-02-12 15:51:27	115	1038

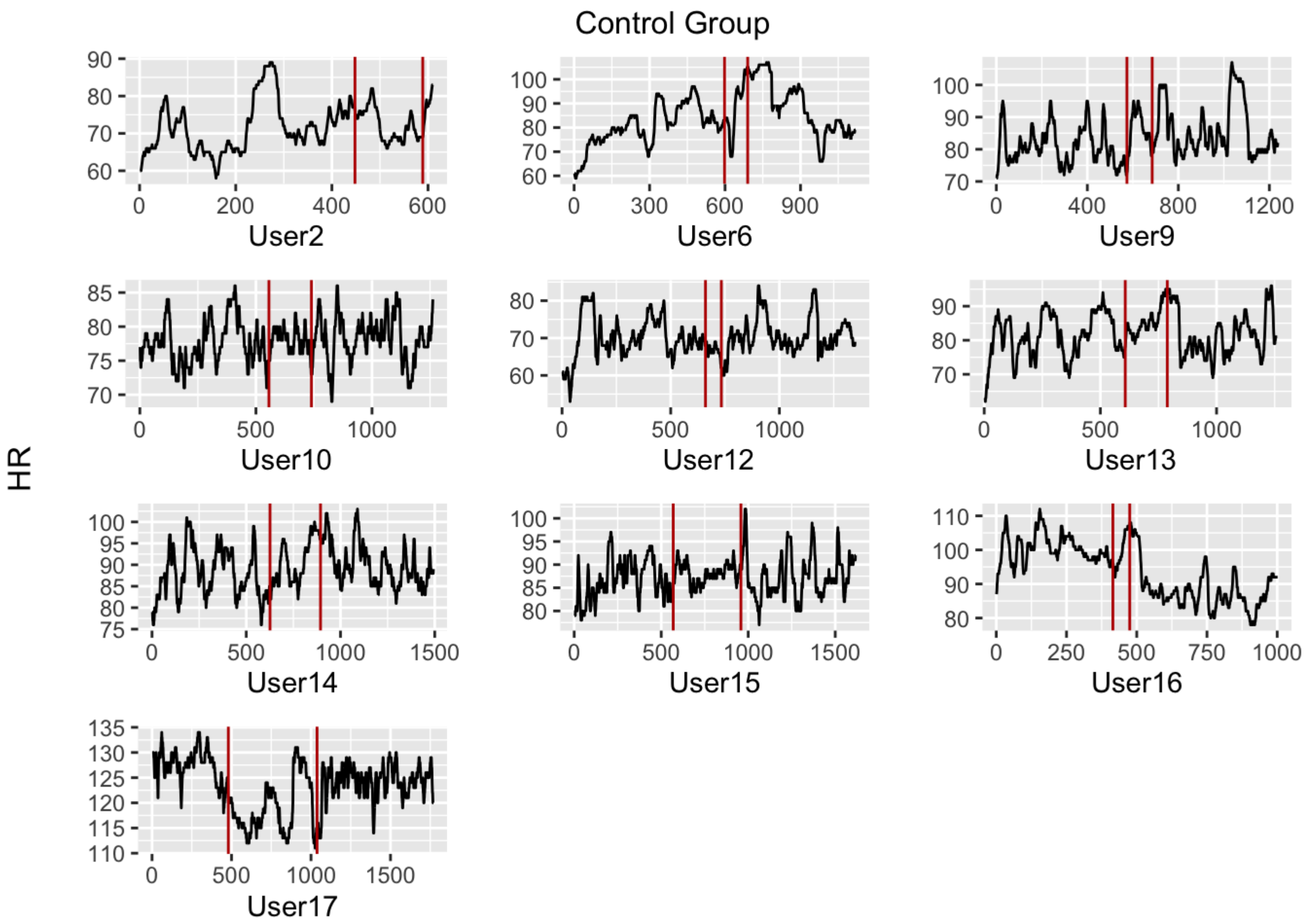
1 row

```
#Figure 5
#plot HR data. Use intercepts from the previous step as checkpoints and identify the area of gameplay.

library(gridExtra)
```

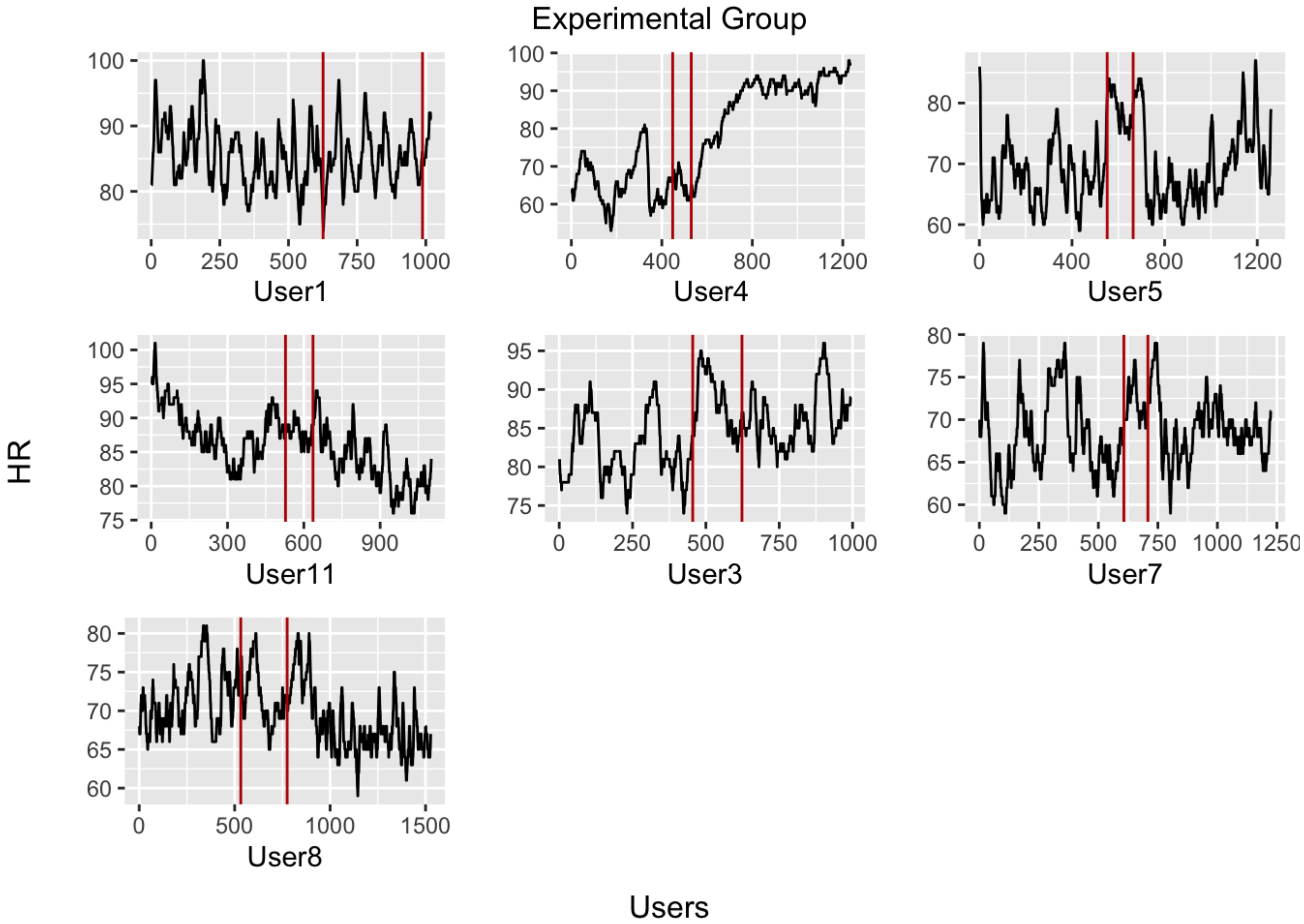
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

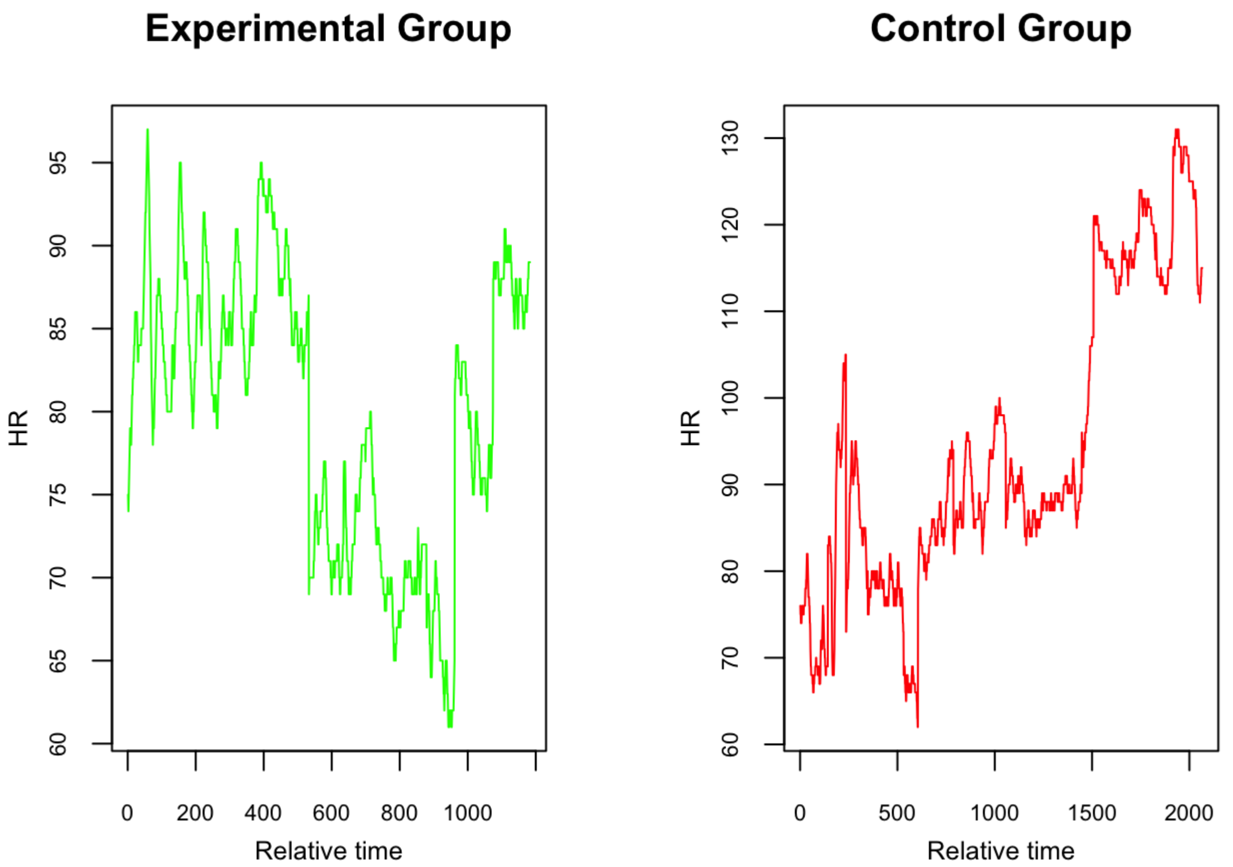
```r
library(ggplot2)
p4<- ggplot(data=user4, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(626,988),colour="#BB0
000") +   labs(x = "User1", y = "")
p5 <- ggplot(data=user5, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(448,589),colour="#BB
0000") +  labs(x = "User2", y = "")
p6 <- ggplot(data=user6, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(455,623),colour="#BB
0000") +  labs(x = "User3", y = "")
p7 <- ggplot(data=user7, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(448,530),colour="#BB
0000") +  labs(x = "User4", y = "")
p8 <- ggplot(data=user8, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(552,664),colour="#BB
0000") +  labs(x = "User5", y = "")
p9 <- ggplot(data=user9, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(598,690),colour="#BB
0000") +  labs(x = "User6", y = "")
p10 <- ggplot(data=user10, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(607,708),colour="#
BB0000") +  labs(x = "User7", y = "")
p11 <- ggplot(data=user11, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(532,775),colour="#
BB0000") +  labs(x = "User8", y = "")
p12 <- ggplot(data=user12, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(574,685),colour="#
BB0000") +  labs(x = "User9", y = "")
p13<- ggplot(data=user13, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(556,739),colour="#B
B0000") +  labs(x = "User10", y = "")
p14 <- ggplot(data=user14, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(528,636),colour="#
BB0000") +  labs(x = "User11", y = "")
p15 <- ggplot(data=user15, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(660,733),colour="#
BB0000") +  labs(x = "User12", y = "")
p16 <- ggplot(data=user16, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(607,788),colour="#
BB0000") +  labs(x = "User13", y = "")
p17 <- ggplot(data=user17, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(626,894),colour="#
BB0000") +  labs(x = "User14", y = "")
p18 <- ggplot(data=user18, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(569,958),colour="#
BB0000") +  labs(x = "User15", y = "")
p19 <- ggplot(data=user19, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(415,475),colour="#
BB0000") +  labs(x = "User16", y = "")
p20 <- ggplot(data=user20, aes (x=counter, y=value)) + geom_line() + geom_vline(xintercept = c(480,1038),colour=
"#BB0000") +  labs(x = "User17", y = "")

grid.arrange(p5,p9,p12,p13,p15,p16,p17,p18,p19,p20, ncol=3,top="Control Group",
             left="HR")
```




```
grid.arrange(p4,p7, p8,p14, p6,p10,p11, ncol=3,top="Experimental Group", bottom="Users",
             left="HR")
```


Experimental Group
HR
User1
User4
User5
User11
User3
User7
User8
Users

```
#Figure 6
#Store the actual HR data between the intercepts for each user.

tsuser4<- user4 %>% filter(counter %in% (626:988))
tsuser5<- user5 %>% filter(counter %in% (448:589))
tsuser6<- user6 %>% filter(counter %in% (455:623))
tsuser7<- user7 %>% filter(counter %in% (448:530))
tsuser8<- user8 %>% filter(counter %in% (552:664))
tsuser9<- user9 %>% filter(counter %in% (598:690))

tsuser10<- user10 %>% filter(counter %in% (607:708))
tsuser11<- user11 %>% filter(counter %in% (532:775))
tsuser12<- user12 %>% filter(counter %in% (574:685))
tsuser13<- user13 %>% filter(counter %in% (556:739))
tsuser14<- user14%>% filter(counter %in% (528:636))
tsuser15<- user15 %>% filter(counter %in% (660:733))
tsuser16<- user16 %>% filter(counter %in% (607:788))
tsuser17<- user17 %>% filter(counter %in% (626:894))
tsuser18<- user18 %>% filter(counter %in% (569:958))
tsuser19<- user19 %>% filter(counter %in% (415:475))
tsuser20<- user20 %>% filter(counter %in% (480:1038))

# Grouping of the stored HR data
experimental <- rbind(tsuser4, tsuser6,tsuser10,tsuser11,tsuser7,tsuser8, tsuser14)
experimental$counter2 <- seq.int(nrow(experimental))

control <- rbind(tsuser5, tsuser9,tsuser12,tsuser13,tsuser15,tsuser16, tsuser17, tsuser18, tsuser19,tsuser20)
control$counter2 <- seq.int(nrow(control))


#Plot data
par(mfrow=c(1,2), mgp=c(2,1,0),mai=c(1,0.4,0.2,0.1),mar = c(4,4,4,2) + 0.1)
plot(experimental$value, type="l",col="green",  ylab ="HR",main= "Experimental Group", xlab ="Relative time",cex.
lab = 0.8,cex.axis= 0.7)
plot(control$value, type="l", col="red", main= "Control Group", ylab ="HR", xlab ="Relative time",cex.lab = 0.8,c
ex.axis= 0.7)
```




```
dev.off()
```

```
## null device
##           1
```

```r
#Table 1
library(MASS)
library(effsize)
library(varhandle)
library(trend)
library(dplyr)
library(gridExtra)

options(scipen=999)

#Let's drop the dates and make some conversions.
data <- data %>% dplyr::select(-StartTime, -EndTime)
data$Avg.speed..m.s.<-as.numeric(gsub(",", ".", gsub("\\.", "", data$Avg.speed..m.s.)))
```

```r
# Fisher's F-test for categorical data
set.seed(882)

gender <- table(data$Group, data$Gender)
print(fisher.test(gender,conf.level = 0.95))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  gender
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.1324252 14.2656768
## sample estimates:
## odds ratio
##   1.310915
```

```r
military <- table(data$Group, data$Military.Service.division.)
print(fisher.test(military,conf.level = 0.95))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  military
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.05747597 11.87273071
## sample estimates:
## odds ratio
##  0.9370952
```

```
# How.often.do.you.park.to.an.underground.parking.space.

parking <- table(data$Group,
                 data$How.often.do.you.park.to.an.underground.parking.space.)
print(fisher.test(parking))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  parking
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.1324252 14.2656768
## sample estimates:
## odds ratio
##   1.310915
```

```
# How.often.do.you.play.video.games
games <- table(data$Group, data$How.often.do.you.play.video.games.)
print(fisher.test(games,alternative='g'))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  games
## p-value = 0.7318
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.1177569       Inf
## sample estimates:
## odds ratio
##  0.8950856
```

```
#How.often.do.you.use.virtual.reality.equipment.
vr <- table(data$Group, data$How.often.do.you.use.virtual.reality.equipment.)
print(fisher.test(vr))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  vr
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.007663633 60.933298385
## sample estimates:
## odds ratio
##  0.6831461
```

```
# TABLE I.  CONTINUES DEMOGRAPHIC AND PHYSIOLOGICAL VARIABLES
#Continuous data

# if F-Test returns p > 0.05 then run t.test with (var.equal = TRUE);
# else F-test returns p < 0.05 (heteroscedasticity) then run (var.equal = FALSE)

#AGE
set.seed(882)
var.test
```

```
## function (x, ...)
## UseMethod("var.test")
## <bytecode: 0x7face3ea6f90>
## <environment: namespace:stats>
```

```
age <- data$Age
group <- data$Group
t.test(age ~ group,var.equal = TRUE,conf.level=0.95)
```

```
##
##  Two Sample t-test
##
## data:  age by group
## t = 0.39768, df = 15, p-value = 0.6965
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.608845  6.723131
## sample estimates:
## mean in group A mean in group B
##        27.20000        26.14286
```

```
#BMI
var.test
```

```
## function (x, ...)
## UseMethod("var.test")
## <bytecode: 0x7face3ea6f90>
## <environment: namespace:stats>
```

```
data$BMI<-as.numeric(gsub(",", ".", gsub("\\.", "", data$BMI)))
BMI <- data$BMI
var.test
```

```
## function (x, ...)
## UseMethod("var.test")
## <bytecode: 0x7face3ea6f90>
## <environment: namespace:stats>
```

```
t.test(BMI ~ group, var.equal = TRUE,conf.level=0.95)
```

```
##
##  Two Sample t-test
##
## data:  BMI by group
## t = -0.99733, df = 15, p-value = 0.3344
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.794766  2.100480
## sample estimates:
## mean in group A mean in group B
##        23.01000        24.85714
```

```
#HR
var.test(control$value, experimental$value)
```

```
##
##  F test to compare two variances
##
## data:  control$value and experimental$value
## F = 4.189, num df = 2065, denom df = 1182, p-value <
## 0.00000000000000022
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  3.783389 4.631560
## sample estimates:
## ratio of variances
##           4.188999
```

```
t.test(control$value, experimental$value, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  control$value and experimental$value
## t = 31.395, df = 3180.1, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.24122 15.00530
## sample estimates:
## mean of x mean of y
##  94.10213  79.97887
```

```
t.test(control$value,experimental$value)
```

```
##
##  Welch Two Sample t-test
##
## data:  control$value and experimental$value
## t = 31.395, df = 3180.1, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.24122 15.00530
## sample estimates:
## mean of x mean of y
##  94.10213  79.97887
```

```
#TABLE II.  GAME PERFORMANCE ANALYTICS VARARAIBLES

Ga<- data%>% filter(Group == "A")
Gb<- data%>% filter(Group == "B")

set.seed(882)
#completion time
comp_a<- Ga$Completion.time..s.
comp_b<- Gb$Completion.time..s.
var.test(comp_a,comp_b) # F Test
```

```
##
##  F test to compare two variances
##
## data:  comp_a and comp_b
## F = 2.4631, num df = 9, denom df = 6, p-value = 0.2849
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   0.4459322 10.6397558
## sample estimates:
## ratio of variances
##           2.463065
```

```
t.test(comp_a,comp_b,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  comp_a and comp_b
## t = 0.55399, df = 15, p-value = 0.5878
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -108.0815  183.9958
## sample estimates:
## mean of x mean of y
##  206.1000  168.1429
```

```
t.test(comp_a,comp_b,alternative= 'l',var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  comp_a and comp_b
## t = 0.55399, df = 15, p-value = 0.7061
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 158.0693
## sample estimates:
## mean of x mean of y
##  206.1000  168.1429
```

```
#walking distance
walk_a<- Ga$Walk.distance..m.
walk_b<- Gb$Walk.distance..m.
var.test(walk_a,walk_b)
```

```
##
##  F test to compare two variances
##
## data:  walk_a and walk_b
## F = 1.8792, num df = 9, denom df = 6, p-value = 0.456
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3402301 8.1177475
## sample estimates:
## ratio of variances
##           1.879229
```

```
t.test(walk_a,walk_b,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  walk_a and walk_b
## t = 0.62471, df = 15, p-value = 0.5415
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -212.9697  389.5697
## sample estimates:
## mean of x mean of y
##     571.3     483.0
```

```
t.test(walk_a,walk_b,alternative= 'l',var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  walk_a and walk_b
## t = 0.62471, df = 15, p-value = 0.7292
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 336.0849
## sample estimates:
## mean of x mean of y
##     571.3     483.0
```

```
#speed
speed_a<- Ga$Avg.speed..m.s.
speed_b<- Gb$Avg.speed..m.s.
var.test(speed_a,speed_b)
```

```
##
##  F test to compare two variances
##
## data:  speed_a and speed_b
## F = 11.635, num df = 9, denom df = 6, p-value = 0.007385
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   2.106551 50.261426
## sample estimates:
## ratio of variances
##           11.63534
```

```
t.test(speed_a, speed_b, var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  speed_a and speed_b
## t = 0.63921, df = 11.095, p-value = 0.5357
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6046947  1.1004090
## sample estimates:
## mean of x mean of y
##  3.255000  3.007143
```

```
t.test(speed_a,speed_b,alternative= 'l',var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  speed_a and speed_b
## t = 0.63921, df = 11.095, p-value = 0.7322
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.9436745
## sample estimates:
## mean of x mean of y
##  3.255000  3.007143
```

```
#stops
stop_a<- Ga$Number.of.stops
stop_b<- Gb$Number.of.stops
var.test(stop_a,stop_b)
```

```
##
##  F test to compare two variances
##
## data:  stop_a and stop_b
## F = 8.8498, num df = 9, denom df = 6, p-value = 0.01528
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   1.602228 38.228479
## sample estimates:
## ratio of variances
##           8.849755
```

```
t.test(stop_a, stop_b, var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  stop_a and stop_b
## t = 1.1475, df = 11.683, p-value = 0.2741
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -24.82213  79.70785
## sample estimates:
## mean of x mean of y
##  53.30000  25.85714
```

```
t.test(stop_a, stop_b,alternative= 'l',var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  stop_a and stop_b
## t = 1.1475, df = 11.683, p-value = 0.8629
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 70.16406
## sample estimates:
## mean of x mean of y
##  53.30000  25.85714
```

```
#rotations
turns_a<- Ga$Number.of.rotations..45.deg.turns.
turns_b<- Gb$Number.of.rotations..45.deg.turns.
var.test(turns_a,turns_b)
```

```
##
##  F test to compare two variances
##
## data:  turns_a and turns_b
## F = 1.2541, num df = 9, denom df = 6, p-value = 0.8098
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2270463 5.4172291
## sample estimates:
## ratio of variances
##           1.254069
```

```
t.test(turns_a, turns_b, var.equal=T)
```

```
## 
##  Two Sample t-test
## 
## data:  turns_a and turns_b
## t = 1.4558, df = 15, p-value = 0.1661
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -62.27264 330.61550
## sample estimates:
## mean of x mean of y 
##  297.6000  163.4286
```

```
t.test(turns_a, turns_b,alternative= 'l',var.equal=T)
```

```
## 
##  Two Sample t-test
## 
## data:  turns_a and turns_b
## t = 1.4558, df = 15, p-value = 0.917
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 295.7405
## sample estimates:
## mean of x mean of y 
##  297.6000  163.4286
```

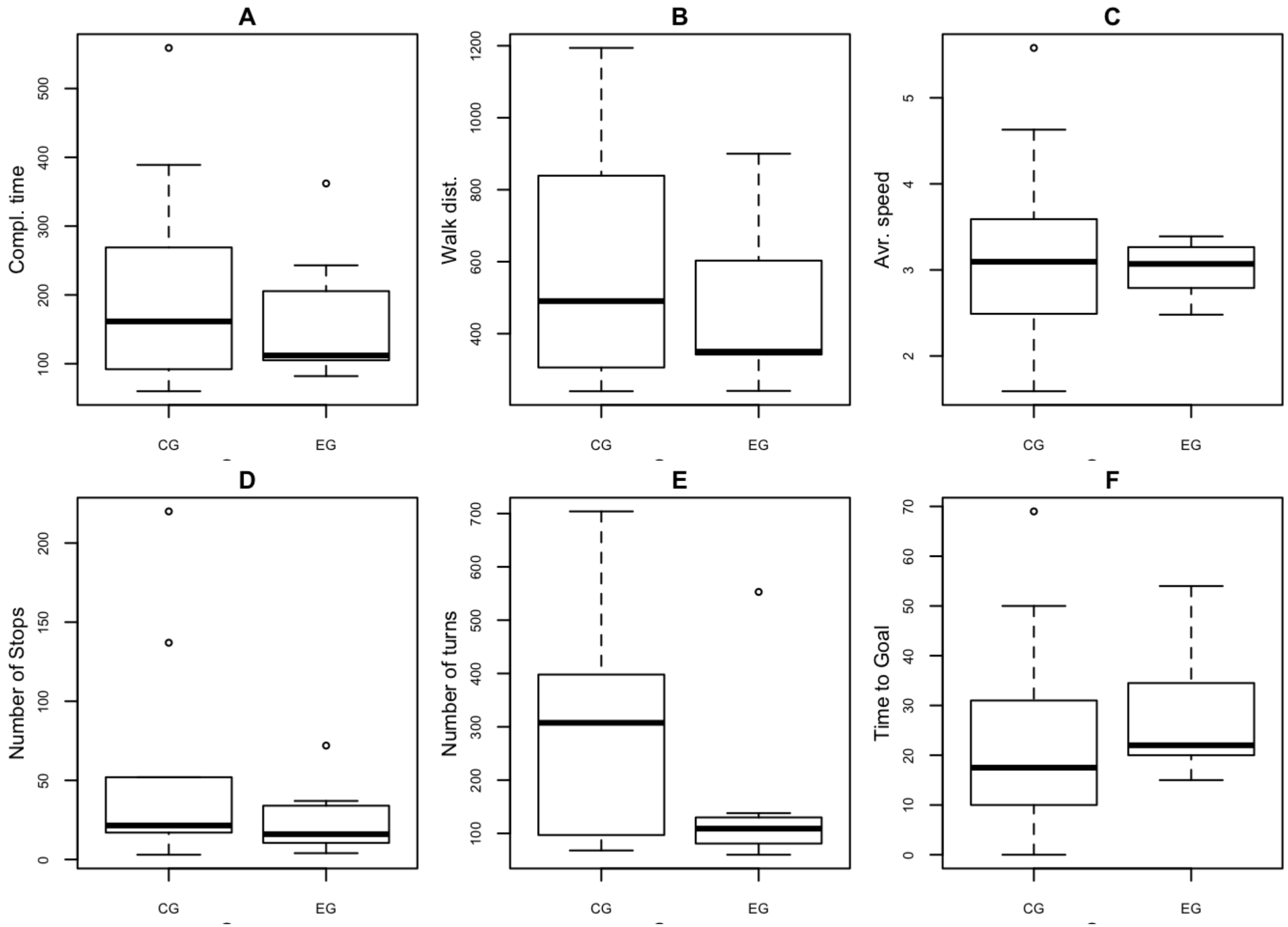
```
#Ambulance found_Goal
goal_a<- Ga$Ambulance.time
goal_b<- Gb$Ambulance.time
var.test(goal_a,goal_b)
```

```
##
##  F test to compare two variances
##
## data:  goal_a and goal_b
## F = 2.3722, num df = 9, denom df = 6, p-value = 0.3053
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   0.4294817 10.2472527
## sample estimates:
## ratio of variances
##           2.372202
```

```
t.test(goal_a, goal_b, var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  goal_a and goal_b
## t = -0.54821, df = 14.964, p-value = 0.5916
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -22.34898  13.20612
## sample estimates:
## mean of x mean of y
##  24.00000  28.57143
```

```
#Figure 4

par(mfrow=c(2,3), mgp=c(2,1,0),mai=c(0.3,0.4,0.2,0.1))

boxplot(comp_a, comp_b, names = c("CG","EG"),
        ylab ="Compl. time", main= "A", xlab ="Group",cex.lab = 1,cex.axis= 0.7)
boxplot(walk_a, walk_b, names = c("CG","EG"),
        ylab ="Walk dist.",main= "B", xlab ="Group",cex.lab = 1,cex.axis= 0.7)
boxplot(speed_a, speed_b, names = c("CG","EG"),
        ylab ="Avr. speed",main= "C", xlab ="Group",cex.lab = 1,cex.axis= 0.7)
boxplot(stop_a, stop_b, names = c("CG","EG"),
        ylab ="Number of Stops", main= "D",xlab ="Group",cex.lab = 1,cex.axis= 0.7)
boxplot(turns_a, turns_b, names = c("CG","EG"),
        ylab ="Number of turns", main= "E",xlab ="Group",cex.lab = 1,cex.axis= 0.7)
boxplot(goal_a, goal_b, names = c("CG","EG"),
        ylab ="Time to Goal", main= "F",xlab ="Group",cex.lab = 1,cex.axis= 0.7)
```




```
dev.off()
```

```
## null device
##           1
```

```
######TABLE III.    PHYSIOLOGICAL TRENDS - Mann-Kendall test for trend
library(trend)
mk.test(experimental$value)
```

```
##
##  Mann-Kendall trend test
##
## data:  experimental$value
## z = -13.094, n = 1183, p-value < 0.00000000000000022
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##                 S              varS               tau
##   -177559.0000000 183877831.0000000        -0.2587377
```

```
mk.test(control$value)
```

```
##
##  Mann-Kendall trend test
##
## data:  control$value
## z = 43.08, n = 2066, p-value < 0.00000000000000022
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##                 S              varS               tau
##   1348362.0000000 979648444.0000000         0.6402402
```