Experimental design and analysis

Introduction to R

https://www.lri.fr/~appert/eval/
The R language

A dynamically typed interpreted language

Many built-in functions and libraries
Types in R

Basic types
Booleans (logical), Numbers (integer or double), Strings (character)

**Vectors** are the fundamental type in R

```r
> a <- c(1,2,5.3,6,-2,4) # numeric vector
> b <- c("one","two","three") # character vector
> c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) # logical vector
> a[c(2,4)] # 2nd and 4th elements of vector a (start index is 1 in R)
[1] 2 6

Matrices: All columns in a matrix must have the same type (numeric, character, etc.) and the same length

```r
> cells <- c(1,26,24,68)
> rnames <- c("R1","R2")
> cnames <- c("C1","C2")
> mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE, dimnames=list(rnames,cnames))
> mymatrix
   C1 C2
R1  1 26
R2 24 68
> mymatrix[1,] # 1st row
   C1 C2
  1 26
> mymatrix[,2] # 2nd column
   R1 R2
  26 68
```
Types in R

**Data frames** are more general than matrices: different columns can have different types (numeric, character, factor, etc.)

```r
> d <- c(1,2,3,4)
> e <- c("red","white","red",NA)
> f <- c(TRUE,TRUE,TRUE,FALSE)
> mydata <- data.frame(d,e,f)
> names(mydata) <- c("ID", "Color", "Hit")
> mydata
   ID Color   Hit
1  1   red  TRUE
2  2 white  TRUE
3  3   red  TRUE
4  4   NA  FALSE
> mydata$ID
[1] 1 2 3 4
```

**Factors** are categorical variables (nominal / ordinal).
Storing data as factors ensures that the statistical models will treat such data correctly. Use `factor` function to create factors

```r
> AvailableTech <- c("none","keyboard","gesture","both")
> AvailableTech
[1] "none"   "keyboard" "gesture" "both"
> typeof(AvailableTech)
[1] "character"
> is.factor(AvailableTech)
[1] FALSE
> AvailableTech <- factor(AvailableTech)
> is.factor(AvailableTech)
[1] TRUE
> AvailableTech
[1] none   keyboard gesture  both
Levels: both gesture keyboard none
(AvailableTech is now a factor that has four categorical levels)
> Difficulty <- ordered(c("easy","medium","hard"))
> Difficulty
[1] easy    medium  hard
Levels: easy < hard < medium
(Difficulty is a factor that has three ordinal levels)
```
Types in R

Check vector type

\textit{is.numeric, is.character, is.factor, etc.}

\begin{Verbatim}
\> d <- c(1,2,3,4)
\> is.numeric(d)
\[1\] TRUE
\> is.factor(d)
[1] FALSE

\> e <- c("red", "white", "red", NA)
\> is.numeric(e)
[1] FALSE
\> is.factor(e)
[1] FALSE
\> is.character(e)
[1] TRUE
\end{Verbatim}

Change vector type

\textit{as.numeric, as.character, etc.}

\begin{Verbatim}
\> num <- c("1","2","3","4")
\> is.numeric(num)
[1] FALSE
\> num <- as.numeric(num)
\> is.numeric(num)
[1] TRUE

\> fac <- factor(c("22", "44", "78"))
\> is.numeric(fac)
[1] FALSE
\> fac <- as.numeric(fac)
\> is.numeric(fac)
[1] TRUE
\end{Verbatim}

\begin{Verbatim}
\> fac
[1] 1 2 3
\end{Verbatim}

\begin{Verbatim}
\# Get a vector of the factor levels:
\> fac <- factor(c("22", "44", "78"))
\> fac <- as.numeric(levels(fac))
\> fac
[1] 22 44 78
\end{Verbatim}
Data frame manipulations

```r
> a <- c(124,235,357,489)
> b <- c(202,306,476,604)
> c <- c(1,2,3,4)
> d <- c("red","white","red","blue")
> e <- c(TRUE,TRUE,TRUE,FALSE)
> mydata <- data.frame(a,b,c,d,e)
> names(mydata) <- c("starttime", "endtime", "ID", "Color", "Hit")

> mydata
   starttime endtime ID  Color  Hit
 1       124     202  1   red  TRUE
 2       235     306  2 white  TRUE
 3       357     476  3   red  TRUE
 4       489     604  4  blue FALSE

> # filter out lines
> mydata <- mydata[!mydata$Color == "white",]

> mydata
   starttime endtime ID  Color  Hit
 1       124     202  1   red  TRUE
 2       357     476  3   red  TRUE
 3       489     604  4  blue FALSE
```

Filter out values (useful for removing practice trials before analyses...)
Data frame manipulations

# add a column
mydata$completiontime <- mydata$endtime - mydata$starttime

Adding a column "completiontime" whose value is endtime-starttime

<table>
<thead>
<tr>
<th>starttime</th>
<th>endtime</th>
<th>ID</th>
<th>Color</th>
<th>Hit</th>
<th>completiontime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124</td>
<td>1</td>
<td>red</td>
<td>TRUE</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>357</td>
<td>3</td>
<td>red</td>
<td>TRUE</td>
<td>119</td>
</tr>
<tr>
<td>4</td>
<td>489</td>
<td>4</td>
<td>blue</td>
<td>FALSE</td>
<td>115</td>
</tr>
</tbody>
</table>

# add a column based on a condition
mydata$difficult <- ifelse(mydata$ID > 2, "yes", "no")

Adding a column "difficult" whose value is yes if ID > 2

<table>
<thead>
<tr>
<th>starttime</th>
<th>endtime</th>
<th>ID</th>
<th>Color</th>
<th>Hit</th>
<th>completiontime</th>
<th>difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124</td>
<td>1</td>
<td>red</td>
<td>TRUE</td>
<td>78</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>357</td>
<td>3</td>
<td>red</td>
<td>TRUE</td>
<td>119</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>489</td>
<td>4</td>
<td>blue</td>
<td>FALSE</td>
<td>115</td>
<td>yes</td>
</tr>
</tbody>
</table>
Create a new project

Import a dataset (e.g., allLogs.csv)

This creates a data frame in the project environment named after the file (e.g., allLogs)

The dataset is displayed in a tabular format in the dataset viewer
Create / Open an R script (e.g., helper_functions.R)
Select functions to add in the environment
Run them
Helper functions

Helper functions used in this class come from
http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_(ggplot2)/#helper-functions

The `summarySE` function treats factor as if they were between-subject factors

The `summarySEwithin` function is more general and treats both between-subject and within-subject factors

For between-subject designs, use the `summarySE` function
For within-subject designs, use the `summarySEwithin` function

```r
> allLogs_summary <- summarySE(allLogs, measurevar="visualSearchTime", groupvars=c("VV","OC"))
> allLogs_summary
       VV     OC  N visualSearchTime        sd        se        ci
1     Color  Large 42         1217.333  167.8740  25.90353  52.31323
2     Color Medium 42         1193.738  133.3084  20.56993  41.54182
3     Color Small 42         1247.357  354.4045  54.68580 110.44021
4 ColorSize  Large 42         2737.214  743.1175 114.66553 231.57173
5 ColorSize Medium 42         2481.952  881.9314 136.08498 274.82918
6 ColorSize Small 42         2393.000 1058.5871 163.34354 329.87897
7      Size  Large 42         1313.119  205.7042  31.74084  64.10193
8      Size Medium 43         1315.907  288.3521  43.97329  88.74168
9      Size  Small 42         1270.762  220.8337  34.07537  68.81661
```

```r
> allLogs_summary <- summarySEwithin(allLogs, measurevar="visualSearchTime", withinvars=c("VV","OC"), idvar="ParticipantID")
> allLogs_summary
       VV     OC  N visualSearchTime visualSearchTime_norm        sd        se        ci
1     Color  Large 42         1217.333              1216.857  181.0708  27.93983  56.42563
2     Color Medium 42         1193.738              1193.262  166.1324  25.63478  51.77049
3     Color Small 42         1247.357              1246.881  417.4093  64.40766 130.07390
4 ColorSize  Large 42         2737.214              2736.738  787.4788 121.51062 245.39567
```

...
R: default summary function

summary provides descriptive statistics of the dataset
counts for character variables (e.g., Technique)
quartiles, min, max, mean for numeric variables (e.g., Time)
RStudio: intro video