Visualization Literacy Analysis

Laura Marusich, Jonathan Bakdash

3/22/2022

Read in data files

```r
#grab files from google drive (only have to do this once)
# source("getFromGoogleDrive.R")

#get the first 6? characters of each data file
#get unique values of these
# this is list of subject ids

raw_file_names <- list.files("AccData")
first_six <- substr(raw_file_names, 1, 6)
sub_ids <- unique(first_six)


# fast_RTs <- data.frame(ParticipantId = character(),
#                         TrialName = character(),
#                         type = character(),
#                         time = numeric()
# )

all_data <- NULL

for (i in 1:length(sub_ids)){

  if (grepl("~$", sub_ids[i], fixed = T)){
    next
  }

  temp_file1 <- read_xlsx(paste0("AccData/", sub_ids[i], "_1.xlsx")) %>%
    slice(1:17) %>%
    select(-starts_with("Order")) %>%
    rename(correct = 8)

  temp_file2 <- read_xlsx(paste0("AccData/", sub_ids[i], "_2.xlsx")) %>%
    slice(1:17) %>%
     select(-starts_with("Order")) %>%
    rename(correct = 8)

  new_temp <- temp_file1 %>%
    bind_cols(temp_file2$correct) %>%
    rename(Correct_1 = correct, Correct_2 = "...9") %>%
    mutate(AnswerRT = TimeToBeginInput - TimeToReadQuestion)
```

```r
  all_data <- all_data %>%
    bind_rows(new_temp)

}

all_data <- all_data %>%
  rename(readRT = TimeToReadQuestion, totalRT = TimeToBeginInput)

#read in trialtype key (I created this from an early version of the previous paper)
trial_type_key <- read.csv("trial_type_key.csv", stringsAsFactors = F)

all_data <- all_data %>%
  mutate(TrialType = trial_type_key$TrialType[match(TrialName, trial_type_key$TrialName)]) %>%
  mutate(TrialType = paste0("Type",TrialType))
```

Basic checks

```r
#how many participants per condition
all_data %>%
  group_by(ParticipantId, Condition) %>%
  summarize(ntrials = n()) %>%
  group_by(Condition) %>%
  summarize(nsubs = n()) %>%
  kable()
```

```
## `summarise()` has grouped output by 'ParticipantId'. You can override using the `.groups`
## argument.
```

Condition	nsubs
VR	50
VR Monitor	39
VR Monitor Stereo	33

```r
#why is the balance so off?
```

Remove outliers based on RT

```r
#removing on trial-by-trial basis


#remove answerRTs below 2000ms first
all_data_remove <- all_data %>%
  filter(AnswerRT >= 2000)
dim(all_data)[1]
```

```
## [1] 2074
```

```r
dim(all_data_remove)[1]
```

```
## [1] 2067
```

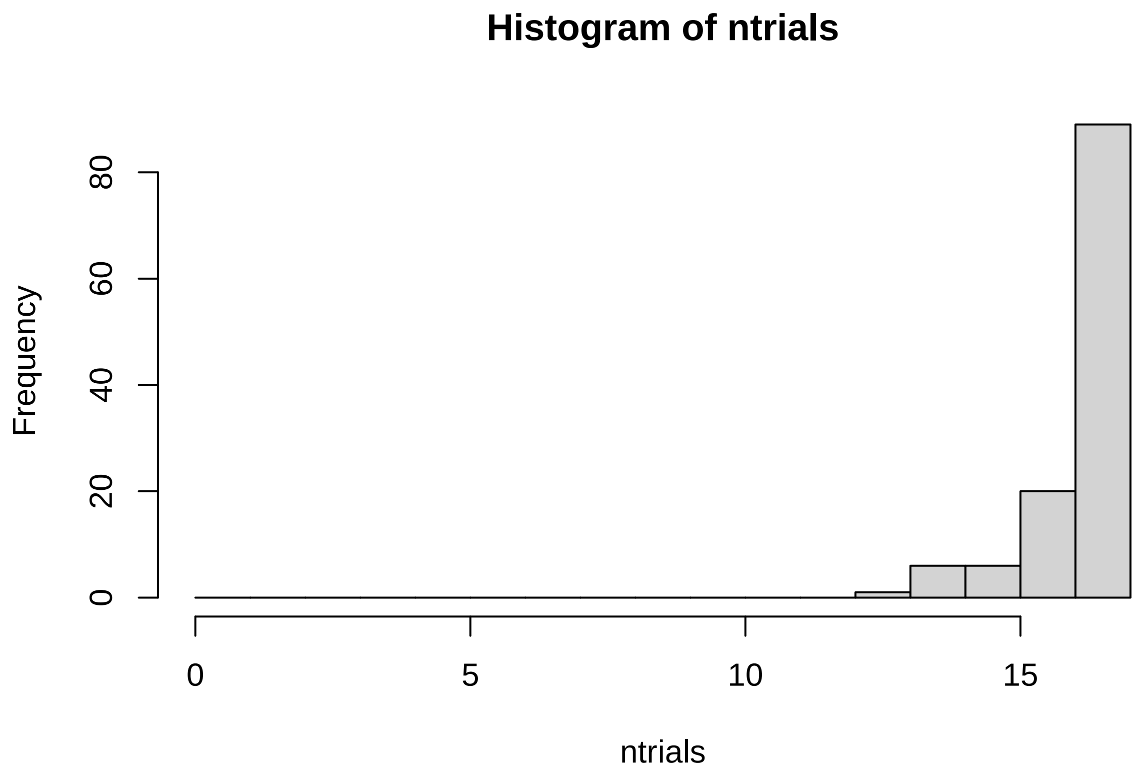
```
#drops 7 trials

rt_data_summary <- all_data %>%
  group_by(TrialName) %>%
  summarize(meanAnswerRT = mean(AnswerRT, na.rm = T),
            sdAnswerRT = sd(AnswerRT, na.rm = T),
            UB = meanAnswerRT + 3*sdAnswerRT,
            LB = meanAnswerRT - 3*sdAnswerRT)
rt_data_summary
```

```
## # A tibble: 17 x 5
##    TrialName     meanAnswerRT sdAnswerRT      UB      LB
##    <chr>                <dbl>      <dbl>   <dbl>   <dbl>
##  1 BarChartQ1          27611.     14762.  71897. -16675.
##  2 BarChartQ2          16921.     13338.  56935. -23093.
##  3 BarChartQ3          15609.      9948.  45452. -14235.
##  4 BarChartQ4          10288.      8978.  37222. -16646.
##  5 LineChartQ1         49185.     29505. 137699. -39329.
##  6 LineChartQ2         40127.     31660. 135107. -54854.
##  7 LineChartQ3         27190.     17504.  79701. -25322.
##  8 LineChartQ4         14779.     16568.  64483. -34925.
##  9 LineChartQ5         27877.     17250.  79628. -23874.
## 10 ScatterplotQ1       32801.     24294. 105682. -40080.
## 11 ScatterplotQ2       29636.     23980. 101576. -42304.
## 12 ScatterplotQ3       45580.     33014. 144623. -53463.
## 13 ScatterplotQ4       35987.     25402. 112194. -40219.
## 14 ScatterplotQ5       59805.     42684. 187859. -68248.
## 15 SurfacePlotQ1       56608.     36042. 164733. -51516.
## 16 SurfacePlotQ2       65110.     50062. 215297. -85076.
## 17 SurfacePlotQ3       48373.     37424. 160646. -63900.
```

```
all_data_no_outliers <- all_data_remove %>%
  group_by(TrialName) %>%
  filter((!(abs(AnswerRT - mean(AnswerRT)) > 3*sd(AnswerRT))))
dim(all_data_no_outliers)[1]
```

```
## [1] 2020
```

```
#drops 47 more trials

all_data_no_outliers %>%
  group_by(ParticipantId) %>%
  summarize(ntrials = n()) %>%
  with(hist(ntrials, breaks = 0:17))
```




```
##maybe consider replacing outliers with means instead of removing them?
```

Compare RTs for conditions and question type (three types: identify, relate, predict)

```
#compare read times (should be no differences of condition)
#compare answer times (potentially a difference)


trial_type_means <- all_data_no_outliers %>%
  group_by(ParticipantId, Condition, TrialType) %>%
  summarize(mean_readRT = mean(readRT),
            mean_answerRT = mean(AnswerRT),
            n = n())
```

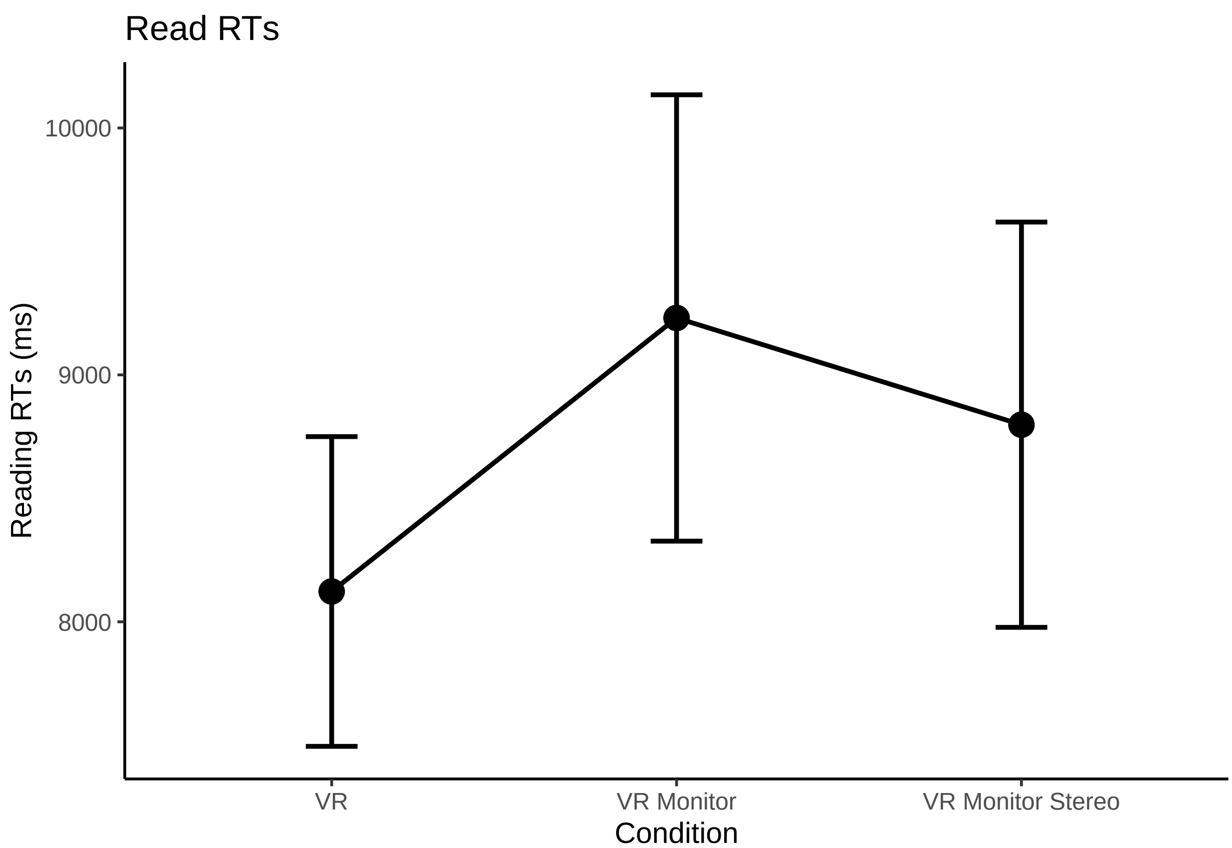
```
## `summarise()` has grouped output by 'ParticipantId', 'Condition'. You can override using
## the `.groups` argument.
```

```
# first, make .csv files in wide format to double check in statview
read_rt_type_wider <- trial_type_means %>%
  select(ParticipantId, Condition, TrialType, mean_readRT) %>%
  pivot_wider(names_from = TrialType,values_from = mean_readRT)
answer_rt_type_wider <- trial_type_means %>%
  select(ParticipantId, Condition, TrialType, mean_answerRT) %>%
  pivot_wider(names_from = TrialType,values_from = mean_answerRT)
write.csv(read_rt_type_wider, file = "readtypeRTs.csv", row.names = F)
```

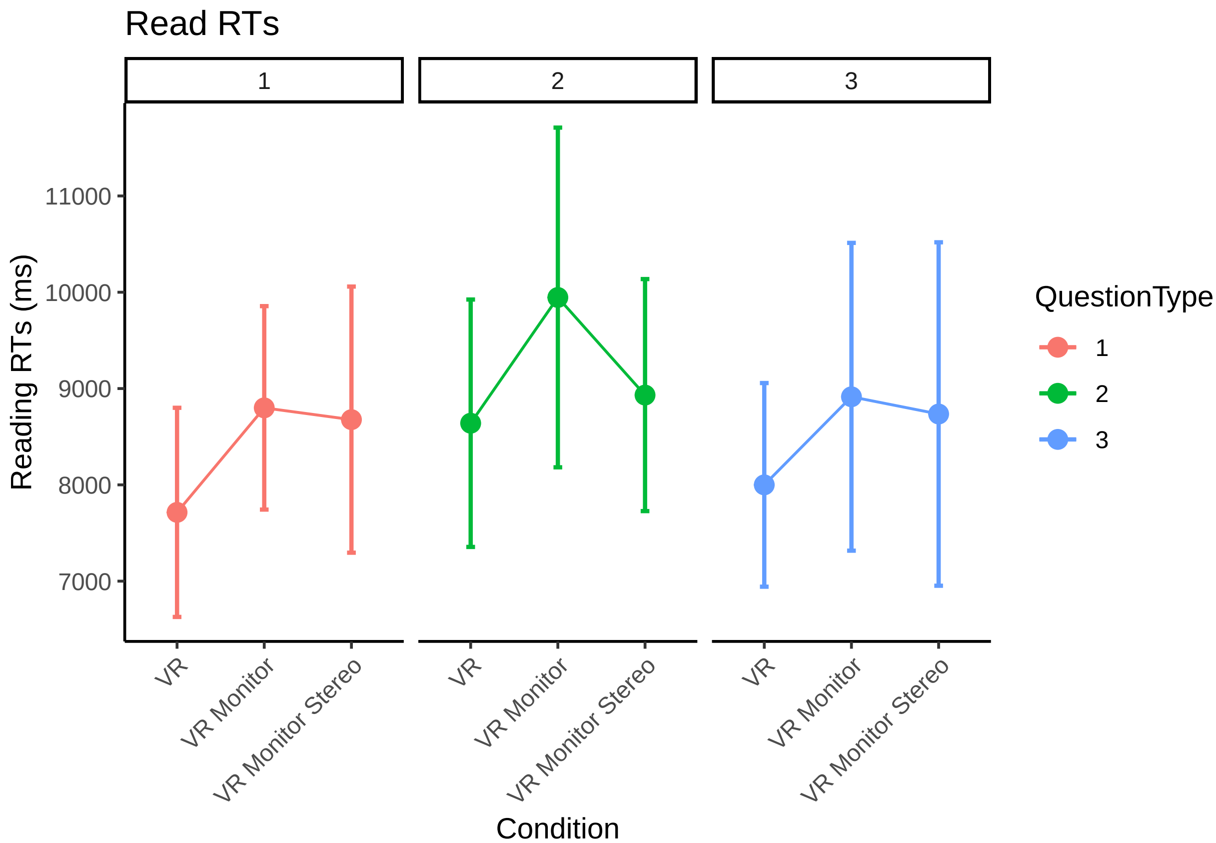
```r
write.csv(answer_rt_type_wider, file = "answertypeRTs.csv", row.names = F)


#### READ RTs ####

#make some plots

#just condition main effect
readplot1 <- all_data_no_outliers %>%
  group_by(ParticipantId, Condition) %>%
  summarize(overallmean = mean(readRT)) %>%
  group_by(Condition) %>%
  summarize(overall_condition_mean = mean(overallmean),
            se = std.error(overallmean),
            n = n(),
            CI = qt(0.975,df=n-1)*se)
```

```
## `summarise()` has grouped output by 'ParticipantId'. You can override using the `.groups`
## argument.
```

```r
 ggplot(readplot1, aes(Condition,
                      overall_condition_mean,
                      group = 1,
                      ymin = overall_condition_mean - CI,
                      ymax = overall_condition_mean + CI)) +
  theme_classic() +
  geom_point(size = 4) +
  geom_errorbar(width = .15, size = 0.85) +
  geom_line(size = 0.85) +
  labs(y = "Reading RTs (ms)", title = "Read RTs")
```




```
#make a little plot using wide format
superbPlot(read_rt_type_wider,
    BSFactors    = "Condition",
    WSFactors    = "QuestionType(3)",
    variables    = c("Type1", "Type2", "Type3"),
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose        = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "QuestionType")
) +
theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
facet_wrap(vars(QuestionType))+
labs(y = "Reading RTs (ms)", title = "Read RTs")
```




```
read_rt_type_anova <- aov_ez(id = "ParticipantId",
                             dv = "mean_readRT",
                             data = trial_type_means,
                              within = "TrialType",
                              between = "Condition",
                             anova_table = list(es = "pes") #might want to double-check these
)
```

```
## Converting to factor: Condition
## Contrasts set to contr.sum for the following variables: Condition
```

```
kable(nice(read_rt_type_anova))
```

Effect	df	MSE	F	pes	p.value
Condition	2, 119	18051925.11	2.28	.037	.107
TrialType	2.00, 237.70	4660052.21	4.30 *	.035	.015
Condition:TrialType	4.00, 237.70	4660052.21	0.52	.009	.723

```
#posthoc test for trial type
pairs(emmeans(read_rt_type_anova, "TrialType"), adjust = "Tukey")
```

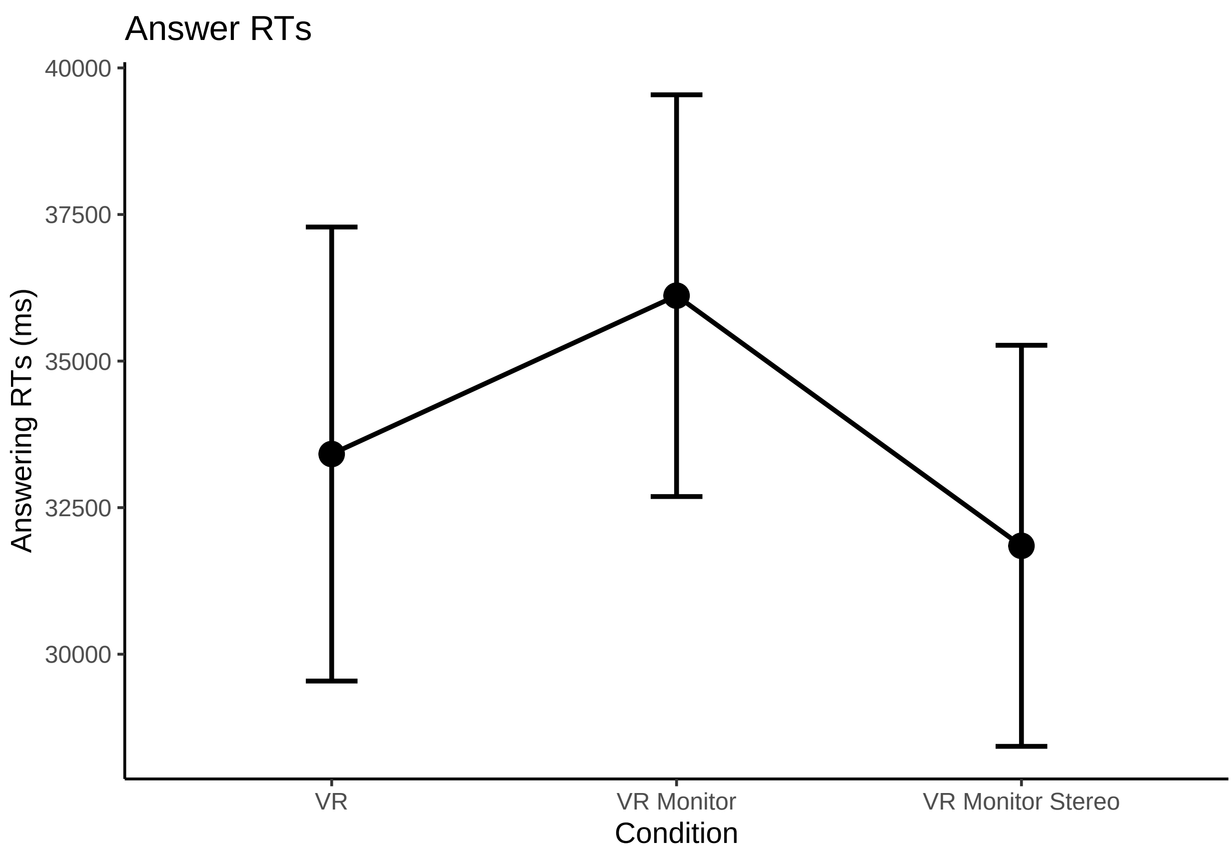
```
##  contrast      estimate  SE  df t.ratio p.value
##  Type1 - Type2     -776 278 119  -2.786  0.0170
##  Type1 - Type3     -153 285 119  -0.537  0.8534
##  Type2 - Type3      623 277 119   2.247  0.0676
##
```

```
## Results are averaged over the levels of: Condition
## P value adjustment: tukey method for comparing a family of 3 estimates
```

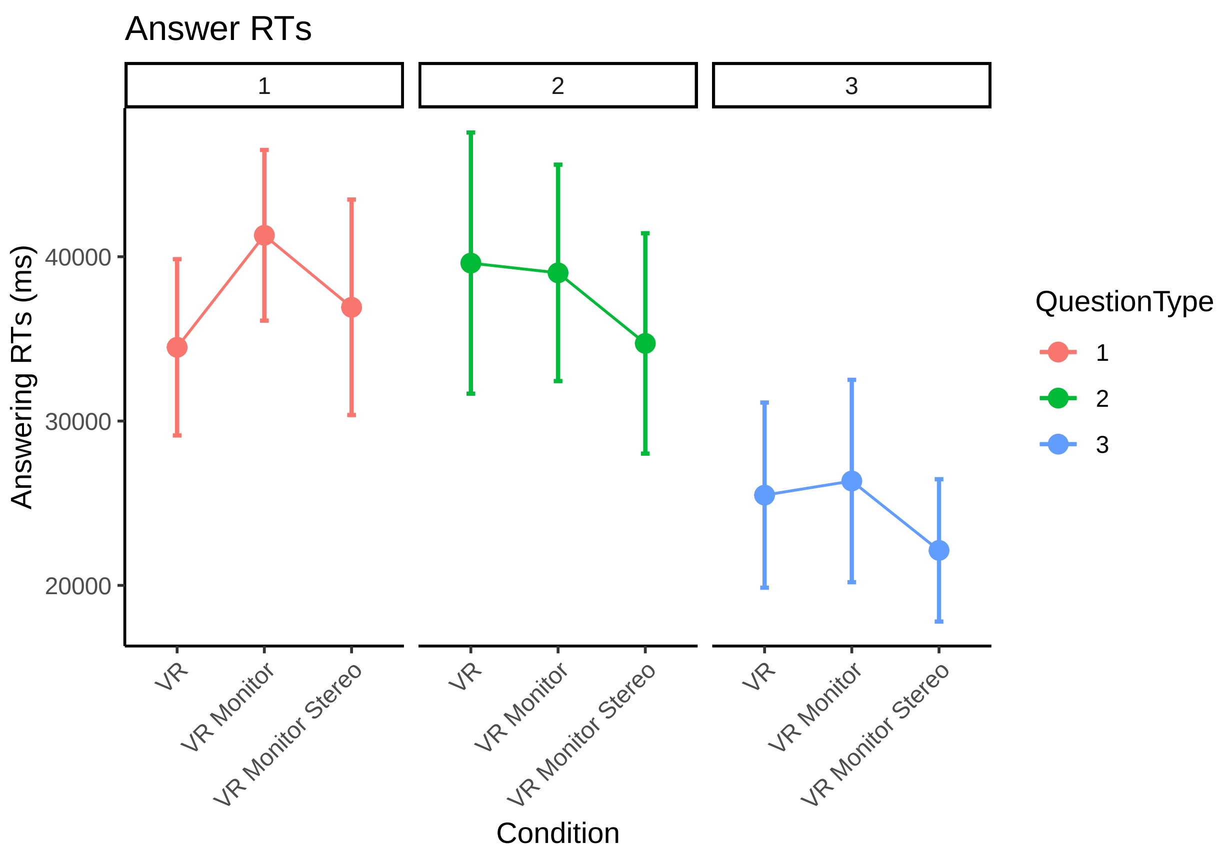
```
#### ANSWER RTs ####

#make some plots

#just condition main effect
answerplot1 <- all_data_no_outliers %>%
   group_by(ParticipantId, Condition) %>%
   summarize(overallmean = mean(AnswerRT)) %>%
   group_by(Condition) %>%
   summarize(overall_condition_mean = mean(overallmean),
             se = std.error(overallmean),
             n = n(),
             CI = qt(0.975,df=n-1)*se)
```

```
## `summarise()` has grouped output by 'ParticipantId'. You can override using the `.groups`
## argument.
```

```
 ggplot(answerplot1, aes(Condition,
                       overall_condition_mean,
                       group = 1,
                       ymin = overall_condition_mean - CI,
                       ymax = overall_condition_mean + CI)) +
   theme_classic() +
   geom_point(size = 4) +
   geom_errorbar(width = .15, size = 0.85) +
   geom_line(size = 0.85) +
   labs(y = "Answering RTs (ms)", title = "Answer RTs")
```




```
#make the little plot
superbPlot(answer_rt_type_wider,
    BSFactors    = "Condition",
    WSFactors    = "QuestionType(3)",
    variables    = c("Type1", "Type2", "Type3"),
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose        = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "QuestionType")
) +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
facet_wrap(vars(QuestionType))+
labs(y = "Answering RTs (ms)", title = "Answer RTs")
```




```r
answer_rt_type_anova <- aov_ez(id = "ParticipantId",
                               dv = "mean_answerRT",
                               data = trial_type_means,
                                within = "TrialType",
                                between = "Condition",
                               anova_table = list(es = "pes") #might want to double-check these
)
```

```
## Converting to factor: Condition
## Contrasts set to contr.sum for the following variables: Condition
```

```r
kable(nice(answer_rt_type_anova))
```

Effect	df	MSE	F	pes	p.value
Condition	2, 119	415942756.23	1.21	.020	.303
TrialType	1.93, 229.53	92112713.90	75.40 ***	.388	<.001
Condition:TrialType	3.86, 229.53	92112713.90	2.52 *	.041	.044

```r
#posthoc test for trial type
pairs(emmeans(answer_rt_type_anova, "TrialType"), adjust = "Tukey")
```

```
##  contrast      estimate   SE  df t.ratio p.value
##  Type1 - Type2     -214 1188 119  -0.180  0.9824
##  Type1 - Type3    12913 1142 119  11.304  <.0001
##  Type2 - Type3    13127 1334 119   9.837  <.0001
##
```

```
## Results are averaged over the levels of: Condition
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
#Question Type 3 much faster than Type 1/Type 2 (this is answering times)

ref <- emmeans(answer_rt_type_anova,~Condition|TrialType)

pairs(ref, adjust = "Tukey")
```

```
## TrialType = Type1:
##  contrast                      estimate   SE  df t.ratio p.value
##  VR - VR Monitor                  -6816 2704 119  -2.521  0.0346
##  VR - VR Monitor Stereo           -2434 2839 119  -0.857  0.6682
##  VR Monitor - VR Monitor Stereo    4382 2994 119   1.464  0.3121
##
## TrialType = Type2:
##  contrast                      estimate   SE  df t.ratio p.value
##  VR - VR Monitor                    595 3542 119   0.168  0.9846
##  VR - VR Monitor Stereo            4887 3718 119   1.314  0.3900
##  VR Monitor - VR Monitor Stereo    4292 3921 119   1.095  0.5192
##
## TrialType = Type3:
##  contrast                      estimate   SE  df t.ratio p.value
##  VR - VR Monitor                   -859 2690 119  -0.319  0.9453
##  VR - VR Monitor Stereo            3364 2824 119   1.191  0.4608
##  VR Monitor - VR Monitor Stereo    4223 2978 119   1.418  0.3350
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
#plot and interaction suggests that the conditions have different effects for different
#question types. posthoc tests indicate a difference between VR and VRMonitor for QType 1
```

Exploratory: Compare RTs for graph type

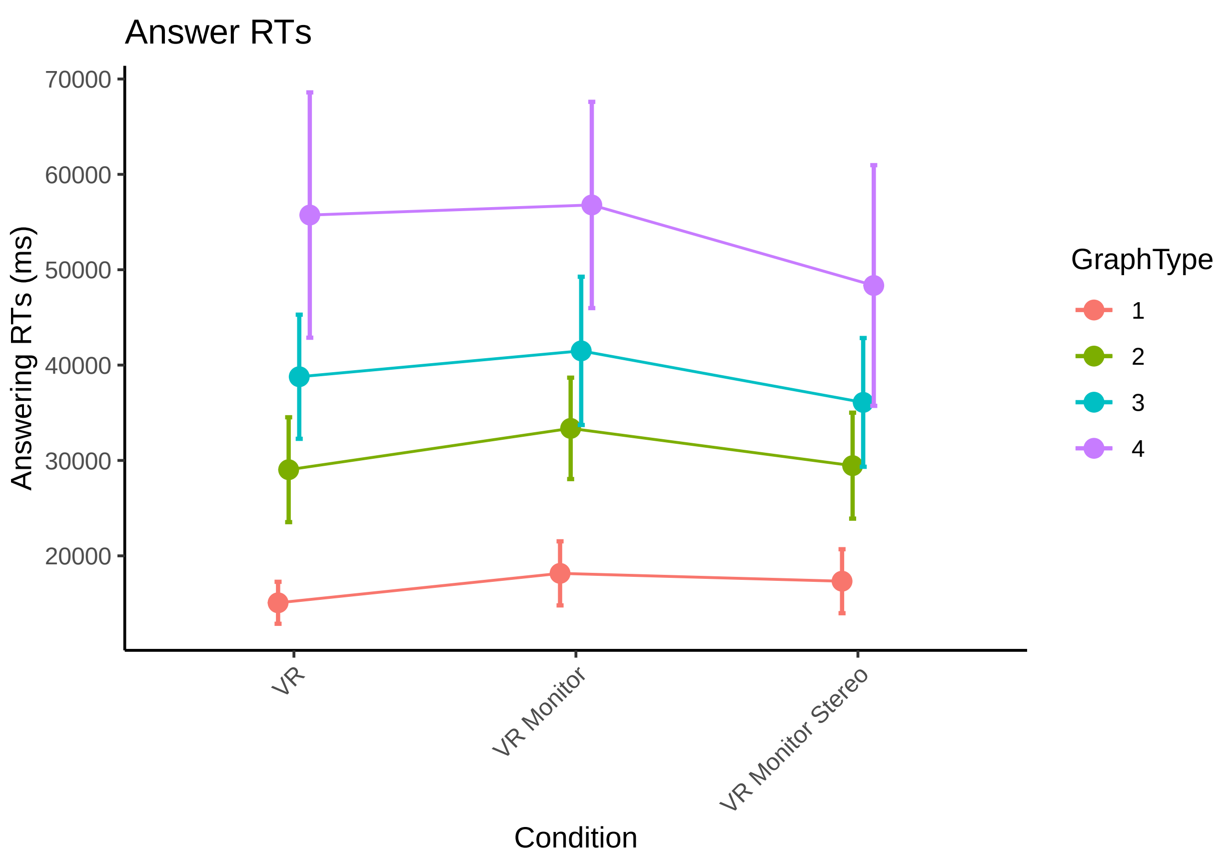
Definite differences in performance for different graph types, but no interaction with condition

```r
all_data_no_outliers <- all_data_no_outliers %>%
  mutate(graph.type = strsplit(TrialName, "Q")[[1]][1])

graph_type_means <- all_data_no_outliers %>%
  group_by(ParticipantId, Condition, graph.type) %>%
  summarize(mean_readRT = mean(readRT),
            mean_answerRT = mean(AnswerRT),
            n = n())
```

```
## `summarise()` has grouped output by 'ParticipantId', 'Condition'. You can override using
## the `.groups` argument.
```

```r
graph_type_wider <- graph_type_means %>%
  select(ParticipantId, Condition, graph.type, mean_answerRT) %>%
  pivot_wider(names_from = graph.type,values_from = mean_answerRT)

superbPlot(graph_type_wider,
    BSFactors   = "Condition",
    WSFactors   = "GraphType(4)",
    variables   = c("BarChart", "LineChart", "Scatterplot", "SurfacePlot"),
```

```r
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose        = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "GraphType")
) +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
# facet_wrap(vars(GraphType))+
labs(y = "Answering RTs (ms)", title = "Answer RTs")
```




```r
answer_rt_graph_type_anova <- aov_ez(id = "ParticipantId",
                             dv = "mean_answerRT",
                             data = graph_type_means,
                              within = "graph.type",
                              between = "Condition",
                             anova_table = list(es = "pes") #might want to double-check these
)
```

```
## Converting to factor: Condition
## Contrasts set to contr.sum for the following variables: Condition
```

```r
kable(nice(answer_rt_graph_type_anova))
```

Effect	df	MSE	F	pes	p.value
Condition	2, 119	595703060.54	1.34	.022	.266
graph.type	1.78, 211.40	347247019.60	136.31 ***	.534	<.001
Condition:graph.type	3.55, 211.40	347247019.60	0.91	.015	.447

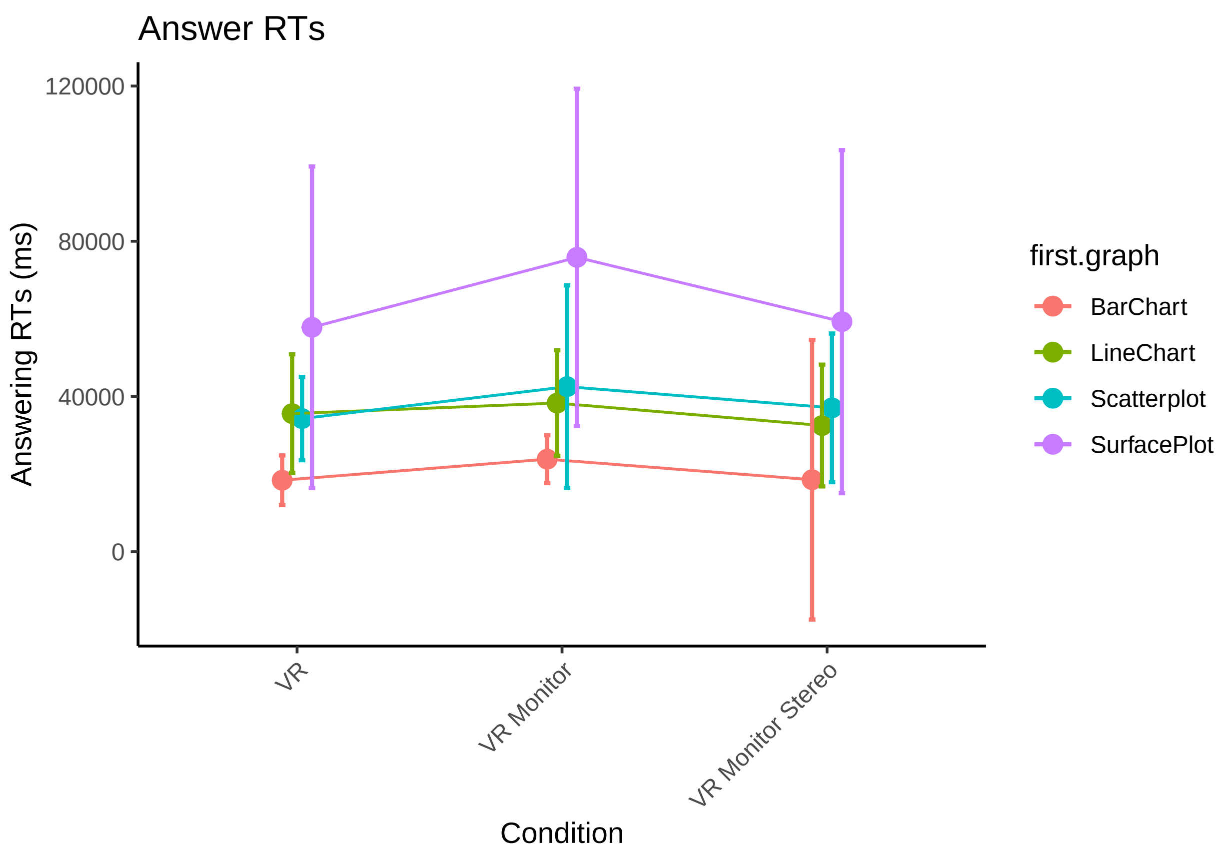
Exploratory: Compare RTs for first graph

```
first_graph <- all_data_no_outliers %>%
  group_by(ParticipantId, Condition) %>%
  mutate(first.graph.type = first(graph.type)) %>%
  filter(graph.type == first.graph.type) %>%
  summarize(
    first.graph = first(first.graph.type),
    mean_RT_first = mean(AnswerRT))
```

```
## `summarise()` has grouped output by 'ParticipantId'. You can override using the `.groups`
## argument.
```

```
superbPlot(first_graph,
    BSFactors   = c("Condition", "first.graph"),
    variables   = "mean_RT_first",
    statistic   = "mean",
    errorbar    = "CI",
    gamma       = 0.95,
    adjustments = list(
        purpose       = "tryon"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "first.graph")
) +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
# facet_wrap(vars(GraphType))+
labs(y = "Answering RTs (ms)", title = "Answer RTs")
```

```
## superb::FYI: The tryon adjustments per measures are  Measure 1: 1.6452 , all compared to 1.4142.
```



Accuracy

```
# do interrater reliability: Fits each trial as independent with two raters
# That is, this ignores participants (complete pooling, which may be problematic)
# 1410 trials from N = 85 participants, not exactly 17 trials per participant
# b/c of excluded trials
irr::icc(select(ungroup(all_data_no_outliers), Correct_1, Correct_2))
```

```
##  Single Score Intraclass Correlation
##
##    Model: oneway
##    Type : consistency
##
##    Subjects = 2020
##      Raters = 2
##      ICC(1) = 0.934
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
## F(2019,2020) = 29.5 , p = 0
##
##  95%-Confidence Interval for ICC Population Values:
##   0.929 < ICC < 0.94
```

```
head(all_data_no_outliers)
```

```
## # A tibble: 6 x 12
```

```
## # Groups:   TrialName [6]
##   Partici~1 Condi~2 Trial~3 Trial~4 readRT totalRT UserI~5 Corre~6 Corre~7 Answe~8 Trial~9
##   <chr>     <chr>     <dbl> <chr>    <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl> <chr>
## 1 AA0176    VR Mon~       1 LineCh~ 16001.  70295. 1.0         0.5     0.5  54294. Type1
## 2 AA0176    VR Mon~       2 LineCh~ 14398.  98640. 6.0         1       1    84243. Type1
## 3 AA0176    VR Mon~       3 LineCh~  9875.  52812. neighb~     0       0    42937. Type2
## 4 AA0176    VR Mon~       4 LineCh~  8780.  13538. Neighb~     1       1     4758. Type3
## 5 AA0176    VR Mon~       5 LineCh~ 18755.  58351. neighb~     0       0    39596. Type3
## 6 AA0176    VR Mon~       6 Scatte~ 12366.  27154. High i~     1       1    14789. Type1
## # ... with 1 more variable: graph.type <chr>, and abbreviated variable names
## #   1: ParticipantId, 2: Condition, 3: TrialNumber, 4: TrialName, 5: UserInput,
## #   6: Correct_1, 7: Correct_2, 8: AnswerRT, 9: TrialType
## # i Use `colnames()` to see all variable names
```

```
# account for non-independence?
  #Data has to be narrow
#https://cran.r-project.org/web/packages/cccrm/index.html
icc.dat.narrow <- all_data_no_outliers %>%
                    pivot_longer(cols = c('Correct_1', 'Correct_2'))

colnames(icc.dat.narrow)[11] <- "Rater"
icc.dat.narrow$Rater <- as.factor(icc.dat.narrow$Rater)

#Longitudinal concordance for two raters
#By all 17 trials (by items)
 icc.rm.trial.name <- ccclon(icc.dat.narrow,
                              "value", "ParticipantId", "TrialName", "Rater")
```

```
## Warning in ccclon(icc.dat.narrow, "value", "ParticipantId", "TrialName", : NAs introduced
## by coercion
```

```
  icc.rm.trial.name
```

```
## CCC estimated by variance components:
##         CCC    LL CI 95%   UL CI 95%      SE CCC
## 0.911213129 0.902891233 0.918852278 0.004067309
```

```
  summary(icc.rm.trial.name)
```

```
## Linear mixed-effects model fit by REML
##   Data: dades
##        AIC     BIC    logLik
##   413.8561 653.087 -168.9281
##
## Random effects:
##  Composite Structure: Blocked
##
##  Block 1: (Intercept)
##  Formula: ~1 | ind
##         (Intercept)
## StdDev:   0.1371422
##
##  Block 2: metCorrect_1, metCorrect_2
##  Formula: ~-1 + met | ind
##  Structure: Multiple of an Identity
##         metCorrect_1 metCorrect_2
```

```
## StdDev: 1.263708e-05 1.263708e-05
##
##  Block 3: timeBarChartQ1, timeBarChartQ2, timeBarChartQ3, timeBarChartQ4, timeLineChartQ1, timeLineCh
##  Formula: ~-1 + time | ind
##  Structure: Multiple of an Identity
##         timeBarChartQ1 timeBarChartQ2 timeBarChartQ3 timeBarChartQ4 timeLineChartQ1
## StdDev:      0.3504545      0.3504545      0.3504545      0.3504545       0.3504545
##         timeLineChartQ2 timeLineChartQ3 timeLineChartQ4 timeLineChartQ5 timeScatterplotQ1
## StdDev:       0.3504545       0.3504545       0.3504545       0.3504545         0.3504545
##         timeScatterplotQ2 timeScatterplotQ3 timeScatterplotQ4 timeScatterplotQ5
## StdDev:         0.3504545         0.3504545         0.3504545         0.3504545
##         timeSurfacePlotQ1 timeSurfacePlotQ2 timeSurfacePlotQ3  Residual
## StdDev:         0.3504545         0.3504545         0.3504545 0.1158222
##
## Fixed effects:  list(form)
##                                   Value  Std.Error   DF    t-value p-value
## (Intercept)                   0.9607012 0.03592034 3885  26.745322  0.0000
## metCorrect_2                 -0.0333333 0.01495258 3885  -2.229269  0.0259
## timeBarChartQ2               -0.0038669 0.04776296 3885  -0.080961  0.9355
## timeBarChartQ3               -0.0104475 0.04776277 3885  -0.218736  0.8269
## timeBarChartQ4                0.0031901 0.04787851 3885   0.066629  0.9469
## timeLineChartQ1              -0.1892704 0.04777230 3885  -3.961927  0.0001
## timeLineChartQ2              -0.2599970 0.04756487 3885  -5.466157  0.0000
## timeLineChartQ3              -0.4347017 0.04787831 3885  -9.079302  0.0000
## timeLineChartQ4              -0.1556339 0.04798588 3885  -3.243326  0.0012
## timeLineChartQ5              -0.5970560 0.04756487 3885 -12.552457  0.0000
## timeScatterplotQ1            -0.0356744 0.04786870 3885  -0.745255  0.4562
## timeScatterplotQ2            -0.4116019 0.04766763 3885  -8.634829  0.0000
## timeScatterplotQ3            -0.2833228 0.04787850 3885  -5.917538  0.0000
## timeScatterplotQ4            -0.5964378 0.04787911 3885 -12.457161  0.0000
## timeScatterplotQ5            -0.4080247 0.04787829 3885  -8.522122  0.0000
## timeSurfacePlotQ1            -0.4436177 0.04777251 3885  -9.286046  0.0000
## timeSurfacePlotQ2            -0.5912336 0.04766784 3885 -12.403196  0.0000
## timeSurfacePlotQ3            -0.6831899 0.04798548 3885 -14.237429  0.0000
## metCorrect_2:timeBarChartQ2     0.0333333 0.02119052 3885   1.573030  0.1158
## metCorrect_2:timeBarChartQ3     0.0249300 0.02119052 3885   1.176468  0.2395
## metCorrect_2:timeBarChartQ4     0.0333333 0.02123556 3885   1.569694  0.1166
## metCorrect_2:timeLineChartQ1    0.0333333 0.02119052 3885   1.573030  0.1158
## metCorrect_2:timeLineChartQ2    0.0333333 0.02110241 3885   1.579599  0.1143
## metCorrect_2:timeLineChartQ3    0.0248588 0.02123556 3885   1.170620  0.2418
## metCorrect_2:timeLineChartQ4    0.0333333 0.02128126 3885   1.566323  0.1174
## metCorrect_2:timeLineChartQ5    0.0168044 0.02110241 3885   0.796327  0.4259
## metCorrect_2:timeScatterplotQ1  0.0418079 0.02123556 3885   1.968769  0.0491
## metCorrect_2:timeScatterplotQ2  0.0333333 0.02114614 3885   1.576332  0.1150
## metCorrect_2:timeScatterplotQ3  0.0502825 0.02123556 3885   2.367844  0.0179
## metCorrect_2:timeScatterplotQ4  0.0248588 0.02123556 3885   1.170620  0.2418
## metCorrect_2:timeScatterplotQ5 -0.0641243 0.02123556 3885  -3.019666  0.0025
## metCorrect_2:timeSurfacePlotQ1  0.0039216 0.02119052 3885   0.185062  0.8532
## metCorrect_2:timeSurfacePlotQ2 -0.0333333 0.02114614 3885  -1.576332  0.1150
## metCorrect_2:timeSurfacePlotQ3  0.0461538 0.02128126 3885   2.168755  0.0302
##  Correlation:
##                                (Intr) mtCr_2 tmBCQ2 tmBCQ3 tmBCQ4 tmLCQ1 tmLCQ2 tmLCQ3
## metCorrect_2                   -0.208
## timeBarChartQ2                 -0.662  0.157
```

```
## timeBarChartQ3                  -0.662  0.157  0.498
## timeBarChartQ4                  -0.661  0.156  0.497  0.497
## timeLineChartQ1                 -0.662  0.156  0.498  0.498  0.497
## timeLineChartQ2                 -0.665  0.157  0.500  0.500  0.499  0.500
## timeLineChartQ3                 -0.661  0.156  0.497  0.497  0.496  0.497  0.499
## timeLineChartQ4                 -0.659  0.156  0.495  0.495  0.495  0.496  0.498  0.494
## timeLineChartQ5                 -0.665  0.157  0.500  0.500  0.499  0.500  0.502  0.499
## timeScatterplotQ1               -0.660  0.156  0.497  0.497  0.496  0.497  0.499  0.496
## timeScatterplotQ2               -0.664  0.157  0.499  0.499  0.498  0.499  0.501  0.498
## timeScatterplotQ3               -0.661  0.156  0.497  0.497  0.496  0.497  0.499  0.496
## timeScatterplotQ4               -0.661  0.156  0.497  0.497  0.496  0.497  0.499  0.496
## timeScatterplotQ5               -0.661  0.156  0.497  0.497  0.496  0.497  0.499  0.496
## timeSurfacePlotQ1               -0.662  0.156  0.498  0.498  0.497  0.498  0.500  0.497
## timeSurfacePlotQ2               -0.664  0.157  0.499  0.499  0.498  0.499  0.501  0.498
## timeSurfacePlotQ3               -0.659  0.156  0.496  0.496  0.494  0.496  0.498  0.495
## metCorrect_2:timeBarChartQ2      0.147 -0.706 -0.222 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeBarChartQ3      0.147 -0.706 -0.110 -0.222 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeBarChartQ4      0.147 -0.704 -0.110 -0.110 -0.222 -0.110 -0.111 -0.110
## metCorrect_2:timeLineChartQ1     0.147 -0.706 -0.110 -0.110 -0.110 -0.222 -0.111 -0.110
## metCorrect_2:timeLineChartQ2     0.147 -0.709 -0.111 -0.111 -0.111 -0.111 -0.222 -0.111
## metCorrect_2:timeLineChartQ3     0.147 -0.704 -0.110 -0.110 -0.110 -0.110 -0.111 -0.222
## metCorrect_2:timeLineChartQ4     0.146 -0.703 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeLineChartQ5     0.147 -0.709 -0.111 -0.111 -0.111 -0.111 -0.111 -0.111
## metCorrect_2:timeScatterplotQ1  0.147 -0.704 -0.110 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeScatterplotQ2  0.147 -0.707 -0.111 -0.111 -0.110 -0.111 -0.111 -0.110
## metCorrect_2:timeScatterplotQ3  0.147 -0.704 -0.110 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeScatterplotQ4  0.147 -0.704 -0.110 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeScatterplotQ5  0.147 -0.704 -0.110 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeSurfacePlotQ1  0.147 -0.706 -0.110 -0.110 -0.110 -0.110 -0.111 -0.110
## metCorrect_2:timeSurfacePlotQ2  0.147 -0.707 -0.111 -0.111 -0.110 -0.111 -0.111 -0.110
## metCorrect_2:timeSurfacePlotQ3  0.146 -0.703 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
##                                 tmLCQ4 tmLCQ5 tmScQ1 tmScQ2 tmScQ3 tmScQ4 tmScQ5 tmSPQ1
## metCorrect_2
## timeBarChartQ2
## timeBarChartQ3
## timeBarChartQ4
## timeLineChartQ1
## timeLineChartQ2
## timeLineChartQ3
## timeLineChartQ4
## timeLineChartQ5                  0.498
## timeScatterplotQ1                0.494  0.499
## timeScatterplotQ2                0.497  0.501  0.498
## timeScatterplotQ3                0.494  0.499  0.496  0.498
## timeScatterplotQ4                0.495  0.499  0.496  0.498  0.496
## timeScatterplotQ5                0.495  0.499  0.496  0.498  0.496  0.496
## timeSurfacePlotQ1                0.496  0.500  0.497  0.499  0.497  0.497  0.497
## timeSurfacePlotQ2                0.497  0.501  0.498  0.500  0.498  0.498  0.498  0.499
## timeSurfacePlotQ3                0.493  0.498  0.494  0.497  0.495  0.494  0.495  0.496
## metCorrect_2:timeBarChartQ2     -0.110 -0.111 -0.110 -0.111 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeBarChartQ3     -0.110 -0.111 -0.110 -0.111 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeBarChartQ4     -0.110 -0.111 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeLineChartQ1    -0.110 -0.111 -0.110 -0.111 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeLineChartQ2    -0.110 -0.111 -0.111 -0.111 -0.111 -0.111 -0.111 -0.111
```

```
## metCorrect_2:timeLineChartQ3   -0.110 -0.111 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeLineChartQ4   -0.222 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeLineChartQ5   -0.110 -0.222 -0.111 -0.111 -0.111 -0.111 -0.111 -0.111
## metCorrect_2:timeScatterplotQ1 -0.110 -0.111 -0.222 -0.110 -0.110 -0.110 -0.110 -0.110
## metCorrect_2:timeScatterplotQ2 -0.110 -0.111 -0.110 -0.222 -0.110 -0.110 -0.110 -0.111
## metCorrect_2:timeScatterplotQ3 -0.110 -0.111 -0.110 -0.110 -0.222 -0.110 -0.110 -0.110
## metCorrect_2:timeScatterplotQ4 -0.110 -0.111 -0.110 -0.110 -0.110 -0.222 -0.110 -0.110
## metCorrect_2:timeScatterplotQ5 -0.110 -0.111 -0.110 -0.110 -0.110 -0.110 -0.222 -0.110
## metCorrect_2:timeSurfacePlotQ1 -0.110 -0.111 -0.110 -0.111 -0.110 -0.110 -0.110 -0.222
## metCorrect_2:timeSurfacePlotQ2 -0.110 -0.111 -0.110 -0.111 -0.110 -0.110 -0.110 -0.111
## metCorrect_2:timeSurfacePlotQ3 -0.109 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110 -0.110
##                                tmSPQ2 tmSPQ3 mC_2:BCQ2 mC_2:BCQ3 mC_2:BCQ4 mC_2:LCQ1
## metCorrect_2
## timeBarChartQ2
## timeBarChartQ3
## timeBarChartQ4
## timeLineChartQ1
## timeLineChartQ2
## timeLineChartQ3
## timeLineChartQ4
## timeLineChartQ5
## timeScatterplotQ1
## timeScatterplotQ2
## timeScatterplotQ3
## timeScatterplotQ4
## timeScatterplotQ5
## timeSurfacePlotQ1
## timeSurfacePlotQ2
## timeSurfacePlotQ3               0.497
## metCorrect_2:timeBarChartQ2    -0.111 -0.110
## metCorrect_2:timeBarChartQ3    -0.111 -0.110  0.498
## metCorrect_2:timeBarChartQ4    -0.110 -0.110  0.497     0.497
## metCorrect_2:timeLineChartQ1   -0.111 -0.110  0.498     0.498     0.497
## metCorrect_2:timeLineChartQ2   -0.111 -0.110  0.500     0.500     0.499     0.500
## metCorrect_2:timeLineChartQ3   -0.110 -0.110  0.497     0.497     0.496     0.497
## metCorrect_2:timeLineChartQ4   -0.110 -0.109  0.496     0.496     0.495     0.496
## metCorrect_2:timeLineChartQ5   -0.111 -0.110  0.500     0.500     0.499     0.500
## metCorrect_2:timeScatterplotQ1 -0.110 -0.110  0.497     0.497     0.496     0.497
## metCorrect_2:timeScatterplotQ2 -0.111 -0.110  0.499     0.499     0.498     0.499
## metCorrect_2:timeScatterplotQ3 -0.110 -0.110  0.497     0.497     0.496     0.497
## metCorrect_2:timeScatterplotQ4 -0.110 -0.110  0.497     0.497     0.496     0.497
## metCorrect_2:timeScatterplotQ5 -0.110 -0.110  0.497     0.497     0.496     0.497
## metCorrect_2:timeSurfacePlotQ1 -0.111 -0.110  0.498     0.498     0.497     0.498
## metCorrect_2:timeSurfacePlotQ2 -0.222 -0.110  0.499     0.499     0.498     0.499
## metCorrect_2:timeSurfacePlotQ3 -0.110 -0.222  0.496     0.496     0.495     0.496
##                                mC_2:LCQ2 mC_2:LCQ3 mC_2:LCQ4 mC_2:LCQ5 mC_2:SQ1 mC_2:SQ2
## metCorrect_2
## timeBarChartQ2
## timeBarChartQ3
## timeBarChartQ4
## timeLineChartQ1
## timeLineChartQ2
## timeLineChartQ3
## timeLineChartQ4
```

```
## timeLineChartQ5
## timeScatterplotQ1
## timeScatterplotQ2
## timeScatterplotQ3
## timeScatterplotQ4
## timeScatterplotQ5
## timeSurfacePlotQ1
## timeSurfacePlotQ2
## timeSurfacePlotQ3
## metCorrect_2:timeBarChartQ2
## metCorrect_2:timeBarChartQ3
## metCorrect_2:timeBarChartQ4
## metCorrect_2:timeLineChartQ1
## metCorrect_2:timeLineChartQ2
## metCorrect_2:timeLineChartQ3    0.499
## metCorrect_2:timeLineChartQ4    0.498     0.495
## metCorrect_2:timeLineChartQ5    0.502     0.499     0.498
## metCorrect_2:timeScatterplotQ1  0.499     0.496     0.495     0.499
## metCorrect_2:timeScatterplotQ2  0.501     0.498     0.497     0.501     0.498
## metCorrect_2:timeScatterplotQ3  0.499     0.496     0.495     0.499     0.496    0.498
## metCorrect_2:timeScatterplotQ4  0.499     0.496     0.495     0.499     0.496    0.498
## metCorrect_2:timeScatterplotQ5  0.499     0.496     0.495     0.499     0.496    0.498
## metCorrect_2:timeSurfacePlotQ1  0.500     0.497     0.496     0.500     0.497    0.499
## metCorrect_2:timeSurfacePlotQ2  0.501     0.498     0.497     0.501     0.498    0.500
## metCorrect_2:timeSurfacePlotQ3  0.498     0.495     0.494     0.498     0.495    0.497
##                                mC_2:SQ3 mC_2:SQ4 mC_2:SQ5 mC_2:SPQ1 mC_2:SPQ2
## metCorrect_2
## timeBarChartQ2
## timeBarChartQ3
## timeBarChartQ4
## timeLineChartQ1
## timeLineChartQ2
## timeLineChartQ3
## timeLineChartQ4
## timeLineChartQ5
## timeScatterplotQ1
## timeScatterplotQ2
## timeScatterplotQ3
## timeScatterplotQ4
## timeScatterplotQ5
## timeSurfacePlotQ1
## timeSurfacePlotQ2
## timeSurfacePlotQ3
## metCorrect_2:timeBarChartQ2
## metCorrect_2:timeBarChartQ3
## metCorrect_2:timeBarChartQ4
## metCorrect_2:timeLineChartQ1
## metCorrect_2:timeLineChartQ2
## metCorrect_2:timeLineChartQ3
## metCorrect_2:timeLineChartQ4
## metCorrect_2:timeLineChartQ5
## metCorrect_2:timeScatterplotQ1
## metCorrect_2:timeScatterplotQ2
## metCorrect_2:timeScatterplotQ3
```

```
## metCorrect_2:timeScatterplotQ4  0.496
## metCorrect_2:timeScatterplotQ5  0.496    0.496
## metCorrect_2:timeSurfacePlotQ1  0.497    0.497    0.497
## metCorrect_2:timeSurfacePlotQ2  0.498    0.498    0.498    0.499
## metCorrect_2:timeSurfacePlotQ3  0.495    0.495    0.495    0.496     0.497
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.61204961 -0.14729994  0.01890441  0.13260950  4.79318660
##
## Number of Observations: 4040
## Number of Groups: 122
##
## CCC estimated by variance components
##         CCC   LL CI 95%   UL CI 95%      SE CCC
## 0.911213129 0.902891233 0.918852278 0.004067309
```

```
#By time using trial numbers (by time)
icc.rm.trial.number <- ccclon(icc.dat.narrow,
                              "value", "ParticipantId", "TrialNumber", "Rater" )
icc.rm.trial.number
```

```
## CCC estimated by variance components:
##         CCC   LL CI 95%   UL CI 95%      SE CCC
## 0.933188983 0.926936472 0.938923387 0.003054415
```

```
summary(icc.rm.trial.number)
```

```
## Linear mixed-effects model fit by REML
##   Data: dades
##        AIC      BIC    logLik
##   1123.783 1363.013 -523.8913
##
## Random effects:
##  Composite Structure: Blocked
##
##  Block 1: (Intercept)
##  Formula: ~1 | ind
##         (Intercept)
## StdDev:   0.1241834
##
##  Block 2: metCorrect_1, metCorrect_2
##  Formula: ~-1 + met | ind
##  Structure: Multiple of an Identity
##         metCorrect_1 metCorrect_2
## StdDev: 9.696363e-06 9.696363e-06
##
##  Block 3: time1, time2, time3, time4, time5, time6, time7, time8, time9, time10, time11, time12, tim
##  Formula: ~-1 + time | ind
##  Structure: Multiple of an Identity
##             time1     time2     time3     time4     time5     time6     time7     time8
## StdDev: 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062
##             time9    time10    time11    time12    time13    time14    time15    time16
## StdDev: 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062 0.4211062
##            time17  Residual
## StdDev: 0.4211062 0.1172055
```

```
##
## Fixed effects:  list(form)
##                          Value  Std.Error   DF   t-value p-value
## (Intercept)          0.7669986 0.04214805 3885 18.197724  0.0000
## metCorrect_2        -0.0172414 0.01538984 3885 -1.120310  0.2627
## time2               -0.2351213 0.05718669 3885 -4.111470  0.0000
## time3               -0.1998475 0.05769169 3885 -3.464060  0.0005
## time4               -0.0837630 0.05695015 3885 -1.470813  0.1414
## time5               -0.2254321 0.05718686 3885 -3.942025  0.0001
## time6               -0.0839962 0.05719564 3885 -1.468577  0.1420
## time7               -0.1155072 0.05695015 3885 -2.028217  0.0426
## time8               -0.1768346 0.05671137 3885 -3.118151  0.0018
## time9                0.0047584 0.05707216 3885  0.083375  0.9336
## time10              -0.0536614 0.05695048 3885 -0.942246  0.3461
## time11              -0.1112609 0.05671137 3885 -1.961879  0.0498
## time12              -0.1821879 0.05769204 3885 -3.157938  0.0016
## time13              -0.0644901 0.05682974 3885 -1.134795  0.2565
## time14              -0.0334820 0.05694998 3885 -0.587920  0.5566
## time15              -0.0200528 0.05707200 3885 -0.351360  0.7253
## time16              -0.0858059 0.05694133 3885 -1.506917  0.1319
## time17              -0.1485382 0.05707216 3885 -2.602638  0.0093
## metCorrect_2:time2   0.0172414 0.02167209 3885  0.795557  0.4263
## metCorrect_2:time3   0.0303993 0.02185976 3885  1.390650  0.1644
## metCorrect_2:time4   0.0047414 0.02158238 3885  0.219688  0.8261
## metCorrect_2:time5   0.0002922 0.02167209 3885  0.013484  0.9892
## metCorrect_2:time6   0.0087668 0.02167209 3885  0.404520  0.6859
## metCorrect_2:time7  -0.0035920 0.02158238 3885 -0.166430  0.8678
## metCorrect_2:time8   0.0090447 0.02149525 3885  0.420775  0.6739
## metCorrect_2:time9  -0.0037670 0.02162691 3885 -0.174182  0.8617
## metCorrect_2:time10 -0.0119253 0.02158238 3885 -0.552547  0.5806
## metCorrect_2:time11 -0.0073488 0.02149525 3885 -0.341879  0.7325
## metCorrect_2:time12 -0.0134604 0.02185976 3885 -0.615760  0.5381
## metCorrect_2:time13  0.0255058 0.02153850 3885  1.184198  0.2364
## metCorrect_2:time14  0.0047414 0.02158238 3885  0.219688  0.8261
## metCorrect_2:time15  0.0130397 0.02162691 3885  0.602939  0.5466
## metCorrect_2:time16  0.0089080 0.02158238 3885  0.412746  0.6798
## metCorrect_2:time17 -0.0205737 0.02162691 3885 -0.951303  0.3415
##  Correlation:
##                     (Intr) mtCr_2 time2  time3  time4  time5  time6  time7  time8  time9
## metCorrect_2        -0.183
## time2               -0.684  0.135
## time3               -0.678  0.133  0.500
## time4               -0.687  0.135  0.507  0.502
## time5               -0.684  0.135  0.505  0.500  0.507
## time6               -0.685  0.135  0.504  0.500  0.507  0.504
## time7               -0.687  0.135  0.507  0.502  0.509  0.507  0.507
## time8               -0.690  0.136  0.509  0.504  0.511  0.509  0.509  0.511
## time9               -0.686  0.135  0.506  0.501  0.508  0.505  0.506  0.508  0.510
## time10              -0.687  0.135  0.507  0.502  0.509  0.507  0.507  0.509  0.511  0.508
## time11              -0.690  0.136  0.509  0.504  0.511  0.509  0.509  0.511  0.513  0.510
## time12              -0.678  0.133  0.500  0.495  0.502  0.500  0.500  0.502  0.504  0.501
## time13              -0.689  0.135  0.508  0.503  0.510  0.508  0.508  0.510  0.512  0.509
## time14              -0.687  0.135  0.507  0.502  0.509  0.507  0.507  0.509  0.511  0.508
## time15              -0.686  0.135  0.505  0.501  0.508  0.505  0.506  0.508  0.510  0.507
```

```
## time16               -0.687  0.135  0.506  0.502  0.509  0.506  0.507  0.509  0.511  0.508
## time17               -0.686  0.135  0.505  0.501  0.508  0.505  0.506  0.508  0.510  0.507
## metCorrect_2:time2    0.130 -0.710 -0.189 -0.095 -0.096 -0.096 -0.096 -0.096 -0.096 -0.096
## metCorrect_2:time3    0.129 -0.704 -0.095 -0.189 -0.095 -0.095 -0.095 -0.095 -0.096 -0.095
## metCorrect_2:time4    0.130 -0.713 -0.096 -0.095 -0.189 -0.096 -0.096 -0.096 -0.097 -0.096
## metCorrect_2:time5    0.130 -0.710 -0.096 -0.095 -0.096 -0.189 -0.096 -0.096 -0.096 -0.096
## metCorrect_2:time6    0.130 -0.710 -0.096 -0.095 -0.096 -0.096 -0.189 -0.096 -0.096 -0.096
## metCorrect_2:time7    0.130 -0.713 -0.096 -0.095 -0.096 -0.096 -0.096 -0.189 -0.097 -0.096
## metCorrect_2:time8    0.131 -0.716 -0.096 -0.095 -0.097 -0.096 -0.096 -0.097 -0.190 -0.097
## metCorrect_2:time9    0.130 -0.712 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.189
## metCorrect_2:time10   0.130 -0.713 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.096
## metCorrect_2:time11   0.131 -0.716 -0.096 -0.095 -0.097 -0.096 -0.096 -0.097 -0.097 -0.097
## metCorrect_2:time12   0.129 -0.704 -0.095 -0.094 -0.095 -0.095 -0.095 -0.095 -0.096 -0.095
## metCorrect_2:time13   0.130 -0.715 -0.096 -0.095 -0.097 -0.096 -0.096 -0.097 -0.097 -0.096
## metCorrect_2:time14   0.130 -0.713 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.096
## metCorrect_2:time15   0.130 -0.712 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.096
## metCorrect_2:time16   0.130 -0.713 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.096
## metCorrect_2:time17   0.130 -0.712 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.097 -0.096
##                      time10 time11 time12 time13 time14 time15 time16 time17 mC_2:2 mC_2:3
## metCorrect_2
## time2
## time3
## time4
## time5
## time6
## time7
## time8
## time9
## time10
## time11                0.511
## time12                0.502  0.504
## time13                0.510  0.512  0.503
## time14                0.509  0.511  0.502  0.510
## time15                0.508  0.510  0.501  0.509  0.508
## time16                0.509  0.511  0.502  0.510  0.509  0.508
## time17                0.508  0.510  0.501  0.509  0.508  0.507  0.508
## metCorrect_2:time2   -0.096 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.096
## metCorrect_2:time3   -0.095 -0.096 -0.094 -0.095 -0.095 -0.095 -0.095 -0.095  0.500
## metCorrect_2:time4   -0.096 -0.097 -0.095 -0.097 -0.096 -0.096 -0.096 -0.096  0.506  0.502
## metCorrect_2:time5   -0.096 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.096  0.504  0.500
## metCorrect_2:time6   -0.096 -0.096 -0.095 -0.096 -0.096 -0.096 -0.096 -0.096  0.504  0.500
## metCorrect_2:time7   -0.096 -0.097 -0.095 -0.097 -0.096 -0.096 -0.096 -0.096  0.506  0.502
## metCorrect_2:time8   -0.097 -0.097 -0.095 -0.097 -0.097 -0.097 -0.097 -0.097  0.508  0.504
## metCorrect_2:time9   -0.096 -0.097 -0.095 -0.096 -0.096 -0.096 -0.096 -0.096  0.505  0.501
## metCorrect_2:time10 -0.189 -0.097 -0.095 -0.097 -0.096 -0.096 -0.096 -0.096  0.506  0.502
## metCorrect_2:time11 -0.097 -0.190 -0.095 -0.097 -0.097 -0.097 -0.097 -0.097  0.508  0.504
## metCorrect_2:time12 -0.095 -0.096 -0.189 -0.095 -0.095 -0.095 -0.095 -0.095  0.500  0.496
## metCorrect_2:time13 -0.097 -0.097 -0.095 -0.190 -0.097 -0.096 -0.097 -0.096  0.507  0.503
## metCorrect_2:time14 -0.096 -0.097 -0.095 -0.097 -0.189 -0.096 -0.096 -0.096  0.506  0.502
## metCorrect_2:time15 -0.096 -0.097 -0.095 -0.096 -0.096 -0.189 -0.096 -0.096  0.505  0.501
## metCorrect_2:time16 -0.096 -0.097 -0.095 -0.097 -0.096 -0.096 -0.190 -0.096  0.506  0.502
## metCorrect_2:time17 -0.096 -0.097 -0.095 -0.096 -0.096 -0.096 -0.096 -0.189  0.505  0.501
##                      mC_2:4 mC_2:5 mC_2:6 mC_2:7 mC_2:8 mC_2:9 mC_2:10 mC_2:11 mC_2:12
## metCorrect_2
```

```
## time2
## time3
## time4
## time5
## time6
## time7
## time8
## time9
## time10
## time11
## time12
## time13
## time14
## time15
## time16
## time17
## metCorrect_2:time2
## metCorrect_2:time3
## metCorrect_2:time4
## metCorrect_2:time5   0.506
## metCorrect_2:time6   0.506  0.504
## metCorrect_2:time7   0.508  0.506  0.506
## metCorrect_2:time8   0.511  0.508  0.508  0.511
## metCorrect_2:time9   0.507  0.505  0.505  0.507  0.509
## metCorrect_2:time10  0.508  0.506  0.506  0.508  0.511  0.507
## metCorrect_2:time11  0.511  0.508  0.508  0.511  0.513  0.509  0.511
## metCorrect_2:time12  0.502  0.500  0.500  0.502  0.504  0.501  0.502   0.504
## metCorrect_2:time13  0.510  0.507  0.507  0.510  0.512  0.508  0.510   0.512   0.503
## metCorrect_2:time14  0.508  0.506  0.506  0.508  0.511  0.507  0.508   0.511   0.502
## metCorrect_2:time15  0.507  0.505  0.505  0.507  0.509  0.506  0.507   0.509   0.501
## metCorrect_2:time16  0.508  0.506  0.506  0.508  0.511  0.507  0.508   0.511   0.502
## metCorrect_2:time17  0.507  0.505  0.505  0.507  0.509  0.506  0.507   0.509   0.501
##                     mC_2:13 mC_2:14 mC_2:15 mC_2:16
## metCorrect_2
## time2
## time3
## time4
## time5
## time6
## time7
## time8
## time9
## time10
## time11
## time12
## time13
## time14
## time15
## time16
## time17
## metCorrect_2:time2
## metCorrect_2:time3
## metCorrect_2:time4
## metCorrect_2:time5
```

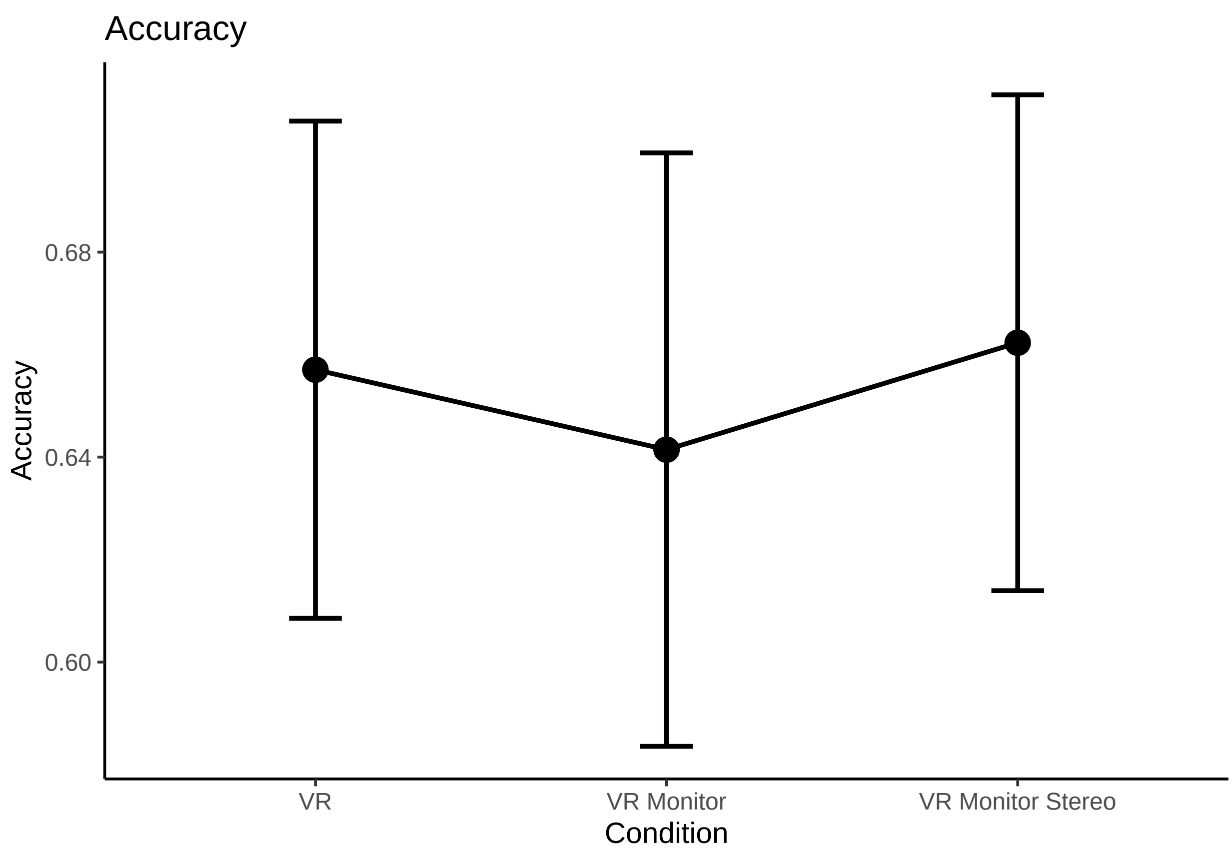
```
## metCorrect_2:time6
## metCorrect_2:time7
## metCorrect_2:time8
## metCorrect_2:time9
## metCorrect_2:time10
## metCorrect_2:time11
## metCorrect_2:time12
## metCorrect_2:time13
## metCorrect_2:time14  0.510
## metCorrect_2:time15  0.508   0.507
## metCorrect_2:time16  0.510   0.508   0.507
## metCorrect_2:time17  0.508   0.507   0.506   0.507
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.42360415 -0.13121079  0.02639034  0.13126188  4.32843966
##
## Number of Observations: 4040
## Number of Groups: 122
##
## CCC estimated by variance components
##         CCC   LL CI 95%   UL CI 95%      SE CCC
## 0.933188983 0.926936472 0.938923387 0.003054415
```

```
# A similar approach
#https://peerj.com/articles/9850/
# test.mod <-
#  lcc(data = hue, subject = "Fruit", resp = "H_mean",
#                  method = "Method", time = "Time", qf = 2, qr = 1)
# summary(test.mod)

# icc.dat.narrow$ParticipantId <- as.factor(icc.dat.narrow$ParticipantId)
# icc.dat.narrow$Rater <- as.factor(icc.dat.narrow$Rater)
#
#
# icc.rm.another <-
# lcc(data = icc.dat.narrow,
#       subject = "ParticipantId",
#       resp = "value",
#       method = "Rater",
#       time = "TrialNumber",
#       qf = 1, #polynomial trends (1 to 3)
#       qr = 0, #random effects (0 is random int only, default)
#       components = TRUE)
#
# #LCC = Longitudinal Concordance Correlation
# #LPC = Longitudinal Pearson Correlation
# #LA  = Longitudinal Accuracy
# icc.rm.another
#
# #Model coefficients near 1 but only the the last one has a correctly
# #estimated model. Maybe because it's more or less flat over TrialNumber?
# lccPlot(icc.rm.another, type = "lpc")
# lccPlot(icc.rm.another, type = "lcc")
```

```r
# iccPlot(icc.rm.another, type = "la")


all_data_no_outliers <- all_data_no_outliers %>%
  rowwise() %>%
  mutate(Correct_Avg = mean(c(Correct_1, Correct_2)))

#just condition main effect
accplot1 <- all_data_no_outliers %>%
   group_by(ParticipantId, Condition) %>%
   summarize(overallmean = mean(Correct_Avg)) %>%
   group_by(Condition) %>%
   summarize(overall_condition_mean = mean(overallmean),
             se = std.error(overallmean),
             n = n(),
             CI = qt(0.975,df=n-1)*se)
```

```
## `summarise()` has grouped output by 'ParticipantId'. You can override using the `.groups`
## argument.
```

```r
 ggplot(accplot1, aes(Condition,
                      overall_condition_mean,
                      group = 1,
                      ymin = overall_condition_mean - CI,
                      ymax = overall_condition_mean + CI)) +
   theme_classic() +
   geom_point(size = 4) +
   geom_errorbar(width = .15, size = 0.85) +
   geom_line(size = 0.85) +
   labs(y = "Accuracy", title = "Accuracy")
```

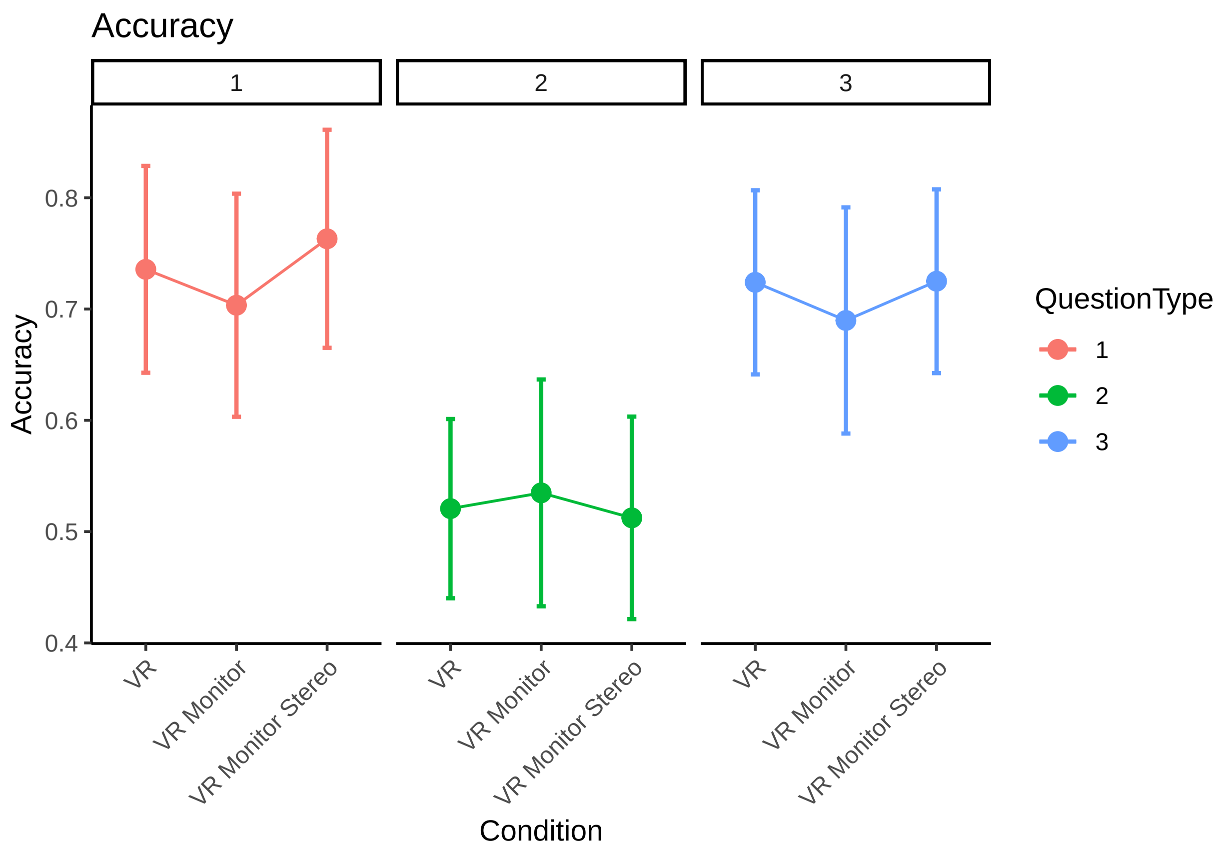



```
trial_type_mean_acc <- all_data_no_outliers %>%
  group_by(ParticipantId, Condition, TrialType) %>%
  summarize(mean_acc = mean(Correct_Avg),
            n = n())
```

```
## `summarise()` has grouped output by 'ParticipantId', 'Condition'. You can override using
## the `.groups` argument.
```

```
acc_type_wider <- trial_type_mean_acc %>%
  select(ParticipantId, Condition, TrialType, mean_acc) %>%
  pivot_wider(names_from = TrialType,values_from = mean_acc)
write.csv(acc_type_wider, file = "typeacc.csv", row.names = F)

#make the little plot
superbPlot(acc_type_wider,
    BSFactors    = "Condition",
    WSFactors    = "QuestionType(3)",
    variables    = c("Type1", "Type2", "Type3"),
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose       = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "QuestionType")
) +
```

```r
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
facet_wrap(vars(QuestionType))+
labs(y = "Accuracy", title = "Accuracy")
```




```r
acc_type_anova <- aov_ez(id = "ParticipantId",
                         dv = "mean_acc",
                         data = trial_type_mean_acc,
                          within = "TrialType",
                          between = "Condition",
                         anova_table = list(es = "pes") #might want to double-check these
)
```

```
## Converting to factor: Condition
## Contrasts set to contr.sum for the following variables: Condition
```

```r
kable(nice(acc_type_anova))
```

Effect	df	MSE	F	pes	p.value
Condition	2, 119	0.08	0.22	.004	.807
TrialType	1.98, 235.08	0.02	66.89 ***	.360	<.001
Condition:TrialType	3.95, 235.08	0.02	0.74	.012	.566

Read in qualtrics data

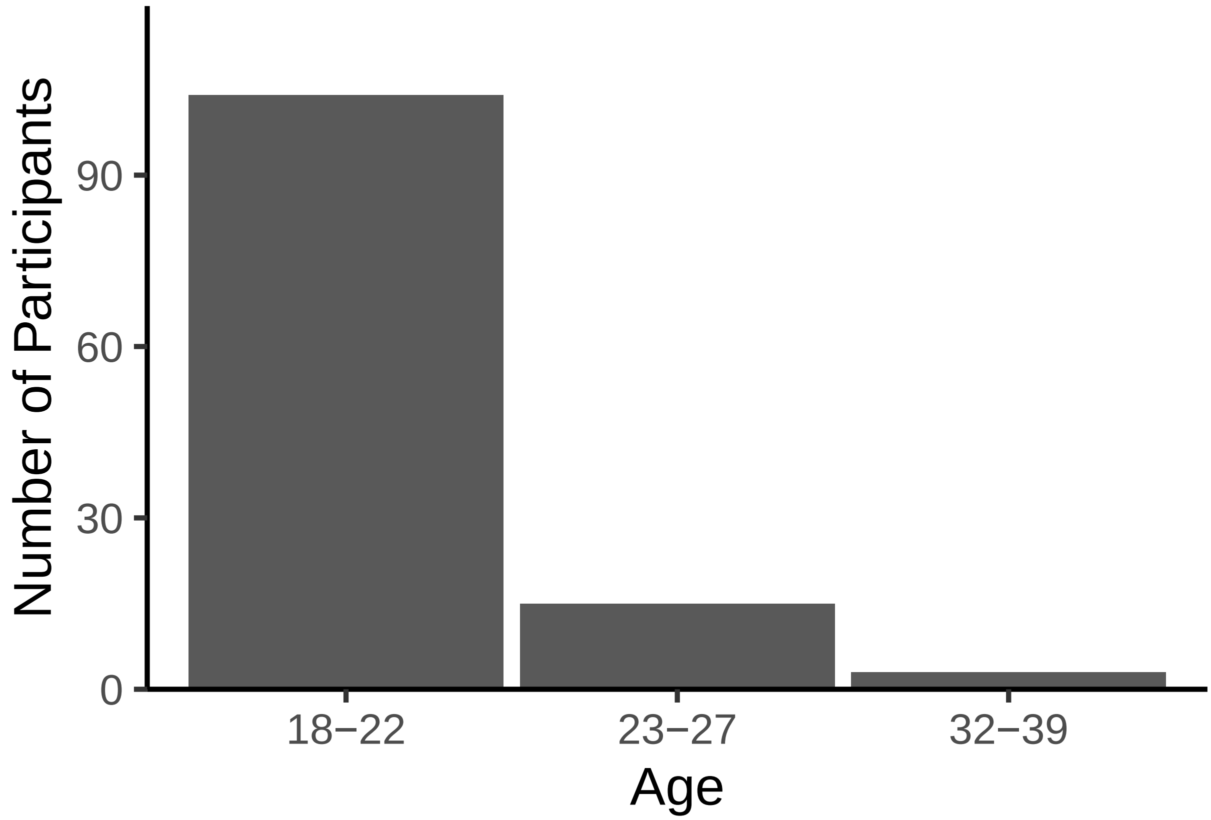
```r
qualtrics <- read_xlsx("04 28 22 Qualtrics Data BNT Coded.xlsx",
                       skip = 3,
                       col_names = c("ParticipantID",
                                     "gender",
                                     "age",
                                     "gaming.exp",
                                     paste0("game.ex", 1:5),
                                     "gaming.hrs.wk",
                                     "fps.exp",
                                     "fps.hrs.wk",
                                     "rts.exp",
                                     "rts.hrs.wk",
                                     "vr.exp",
                                     paste0("vr.times.",
                                            c("","research","gaming","train","dev","other")),
                                      paste0("spes", 1:8),
                                     paste0(rep(c("bnt", "bnt.corr"),4),rep(1:4, each = 2)),
                                     "bnt.tot.corr"))


#note, i think there's a typo in a participant ID. double check at some point
qualtrics <- qualtrics %>%
  mutate(ParticipantID = ifelse(ParticipantID == "U05531", "U05311", ParticipantID))

#remove participants who aren't in behavioral dataset
qualtrics <- qualtrics %>%
  mutate(ParticipantID = toupper(ParticipantID)) %>%
  filter(ParticipantID %in% toupper(all_data_no_outliers$ParticipantId))
```

Summarize qualtrics data

```r
### Age
age_dat <- qualtrics %>%
  group_by(age) %>%
  summarize(n = n())

kable(age_dat)
```

age	n
18-22	104
23-27	15
32-39	3

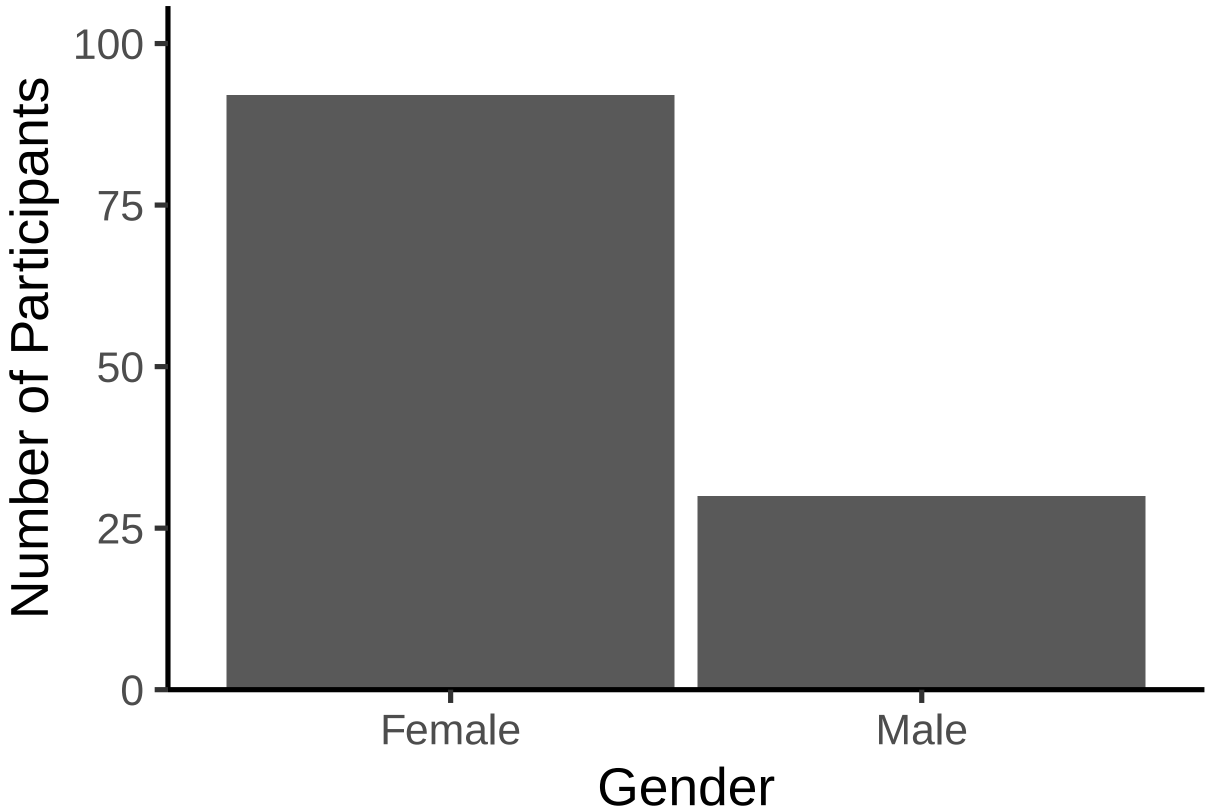
```r
ggplot(age_dat, aes(age, n)) +
  theme_classic(base_size = 20) +
  geom_col(width = .95) +
  labs(y = "Number of Participants", x = "Age") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```




```r
### Gender
gender_dat <- qualtrics %>%
  group_by(gender) %>%
  summarize(n = n())

kable(gender_dat)
```

gender	n
Female	92
Male	30

```r
ggplot(gender_dat, aes(gender, n)) +
  theme_classic(base_size = 20) +
  geom_col(width = .95) +
  labs(y = "Number of Participants", x = "Gender") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```



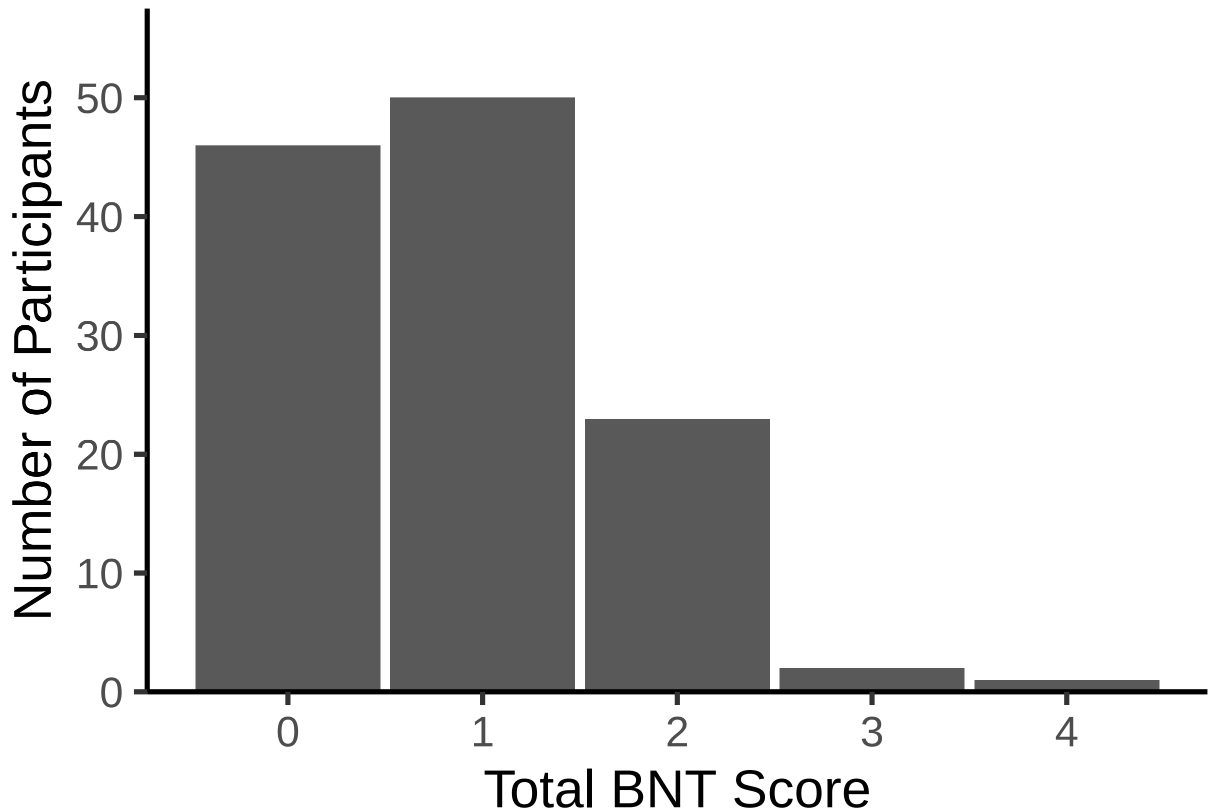

```
### Gaming experience
qualtrics %>%
  group_by(gaming.exp) %>%
  summarize(n = n()) %>%
  kable()
```

gaming.exp	n
No	12
Yes	110

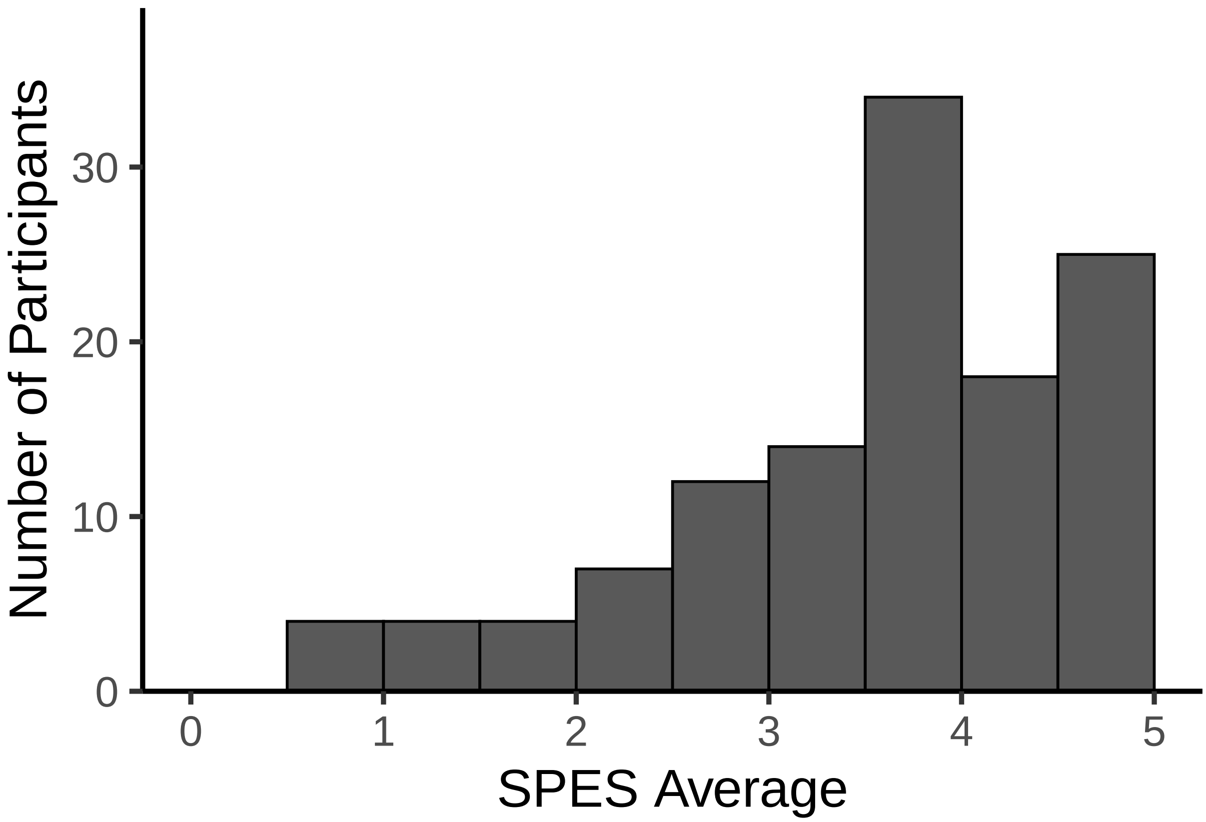
```
### BNT performance
bnt_dat <- qualtrics %>%
  group_by(bnt.tot.corr) %>%
  summarize(n = n())
kable(bnt_dat)
```

bnt.tot.corr	n
0	46
1	50
2	23
3	2
4	1

```r
ggplot(bnt_dat, aes(bnt.tot.corr, n)) +
  theme_classic(base_size = 20) +
  geom_col(width = .95) +
  labs(y = "Number of Participants", x = "Total BNT Score") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```




```r
## SPES responses
## take the average? not sure how to score this
spes_key <- c("I fully disagree", "I somewhat disagree", "I neither disagree nor agree",
              "I somewhat agree", "I fully agree")
spes_dat <- qualtrics %>%
  mutate(across(starts_with("spes"), ~as.numeric(fct_relevel(as_factor(.), spes_key)))) %>%
  mutate(spes.avg = rowSums(across(starts_with("spes")))/8) %>%
  select(ParticipantID, spes.avg)

ggplot(spes_dat, aes(spes.avg)) +
  theme_classic(base_size = 20) +
  geom_histogram(breaks = seq(0,5,by=.5), color = "black") +
  labs(y = "Number of Participants", x = "SPES Average") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```



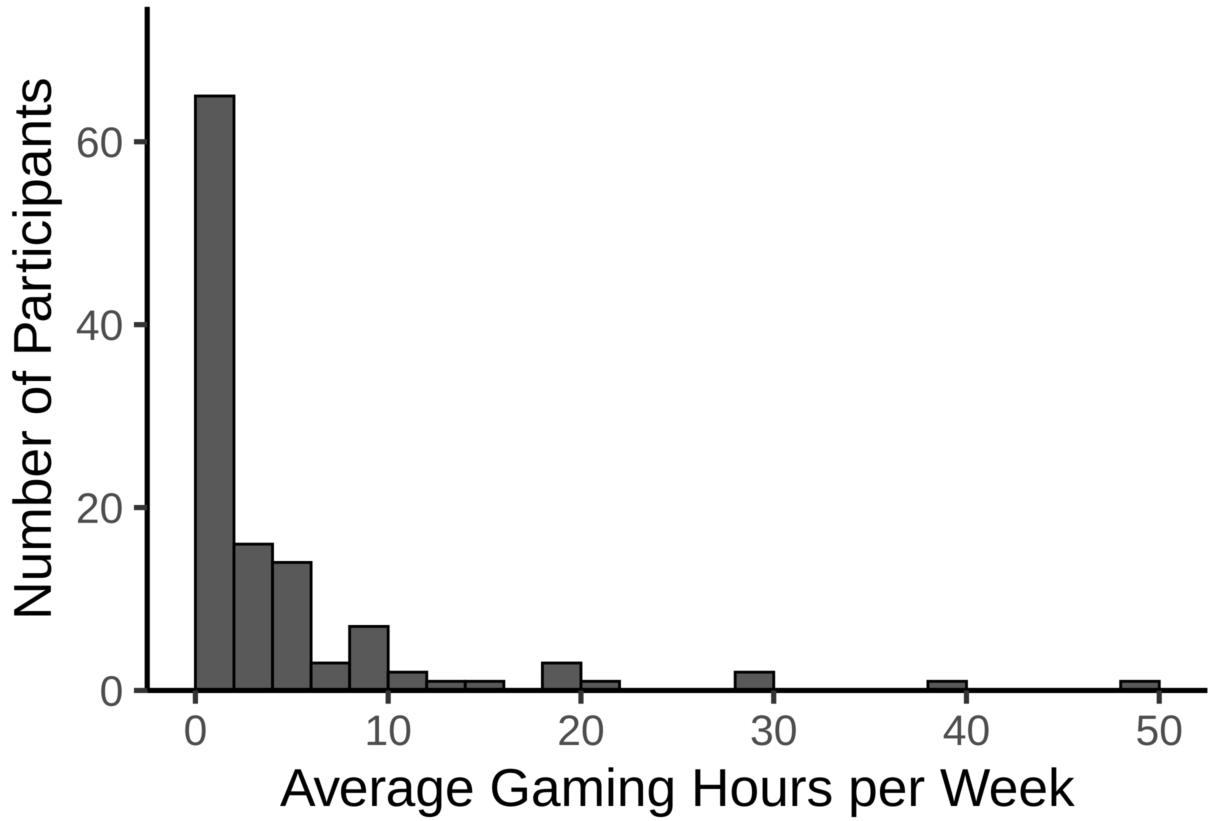

```r
### Video game experience
# this was entered as a free response, and requires some recoding

qualtrics %>%
  select(gaming.hrs.wk) %>%
  unique() %>%
  print(n = Inf)
```

```
## # A tibble: 40 x 1
##    gaming.hrs.wk
##    <chr>
##  1 <NA>
##  2 4
##  3 3
##  4 20
##  5 5
##  6 6
##  7 0
##  8 44291
##  9 1
## 10 50
## 11 10
## 12 21
## 13 2
## 14 No. None.
## 15 less than 30 mins
```

```
## 16 1 2 hours
## 17 12
## 18 less than 2
## 19 7
## 20 1 hour
## 21 rarely
## 22 44481
## 23 2 hours at most
## 24 14
## 25 16
## 26 8
## 27 2.5
## 28 I do not currently play games
## 29 none
## 30 40+
## 31 1 or 2
## 32 30
## 33 3 hours
## 34 AN HOUR OR 2
## 35 0-1
## 36 3.5
## 37 44563
## 38 not recently
## 39 I have not played games recently. It's been more than 5 months
## 40 44659
```

```
game_hrs <- qualtrics %>%
  mutate(gaming.hrs.wk = replace_na(gaming.hrs.wk, "0.0")) %>%
  mutate(gaming.hrs.wk = recode(gaming.hrs.wk,
                                '40+' = '40.0',
                                '44291.0' = '4.5',
                                '44481.0' = '11',
                                '44659.0' = '6.0',
                                '3 hours' = '3.0',
                                '2 hours at most' = '2.0',
                                '1 2 hours' = '1.5',
                                '1 or 2' = '1.5',
                                'AN HOUR OR 2' = '1.5',
                                'less than 2' = '1.5',
                                '44563.0' = '1.5',
                                '1 hour' = '1.0',
                                '0-1' = '0.5',
                                "less than 30 mins" = '0.25',
                                'No. None.' = '0.0',
                                'none' = '0.0',
                                'rarely' = '0.0',
                                "not recently" = '0.0',
                                "I do not currently play games" = '0.0',
                                "I have not played games recently. It's been more than 5 months" = '0.0
                                )) %>%
  mutate(gaming.hrs.wk = as.numeric(gaming.hrs.wk))

ggplot(game_hrs, aes(gaming.hrs.wk)) +
  theme_classic(base_size = 20) +
```

```
  geom_histogram(breaks = seq(0,50,by=2), color = "black") +
  labs(y = "Number of Participants", x = "Average Gaming Hours per Week") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```

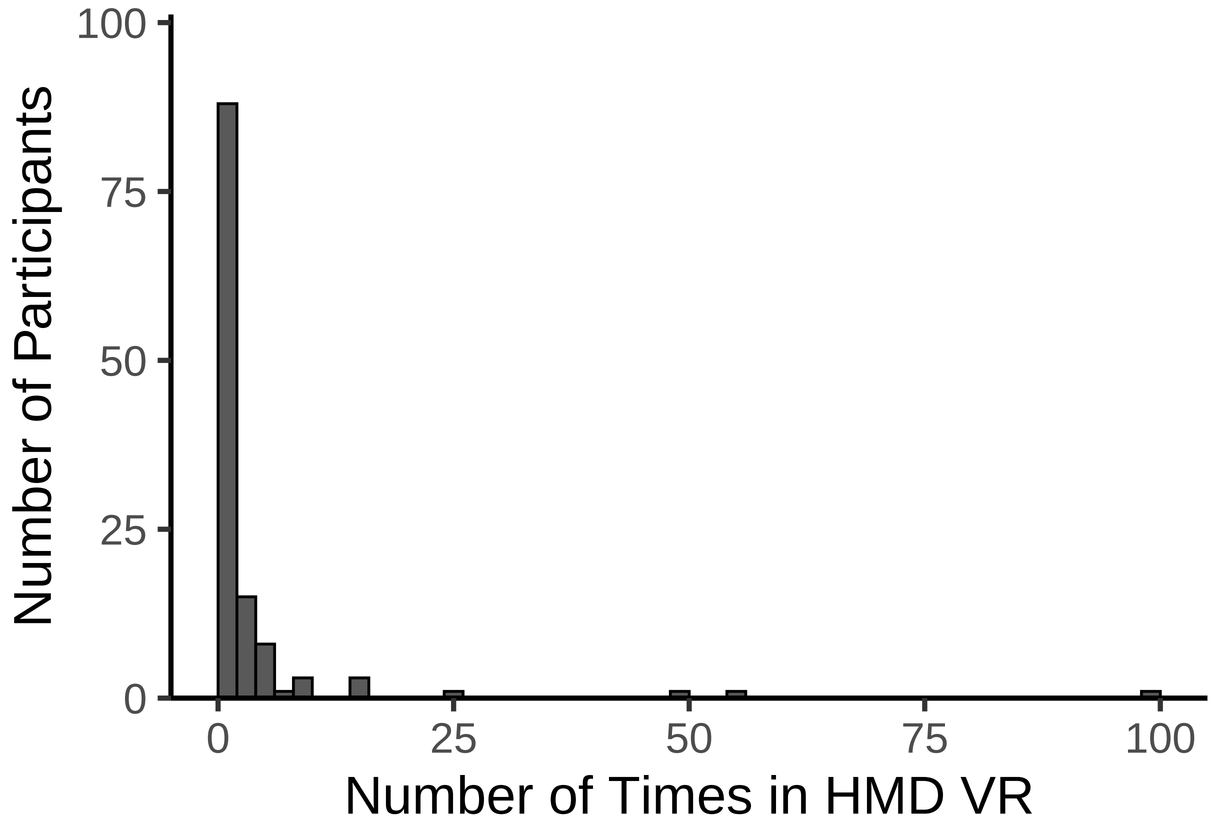



```
### VR Experience
qualtrics %>%
  select(vr.times.) %>%
  unique() %>%
  print(n = Inf)
```

```
## # A tibble: 15 x 1
##    vr.times.
##        <dbl>
##  1        NA
##  2         1
##  3         2
##  4         6
##  5         3
##  6        25
##  7         9
##  8        10
##  9         4
## 10        15
## 11         5
## 12       100
## 13        50
```

```
## 14          7
## 15         55
```

```
vr_times <- qualtrics %>%
  mutate(vr.times. = replace_na(vr.times., 0))

ggplot(vr_times, aes(vr.times.)) +
  theme_classic(base_size = 20) +
  geom_histogram(breaks = seq(0,100,by=2), color = "black") +
  labs(y = "Number of Participants", x = "Number of Times in HMD VR") +
  scale_y_continuous(expand=expansion(mult=c(0,0.15)))
```

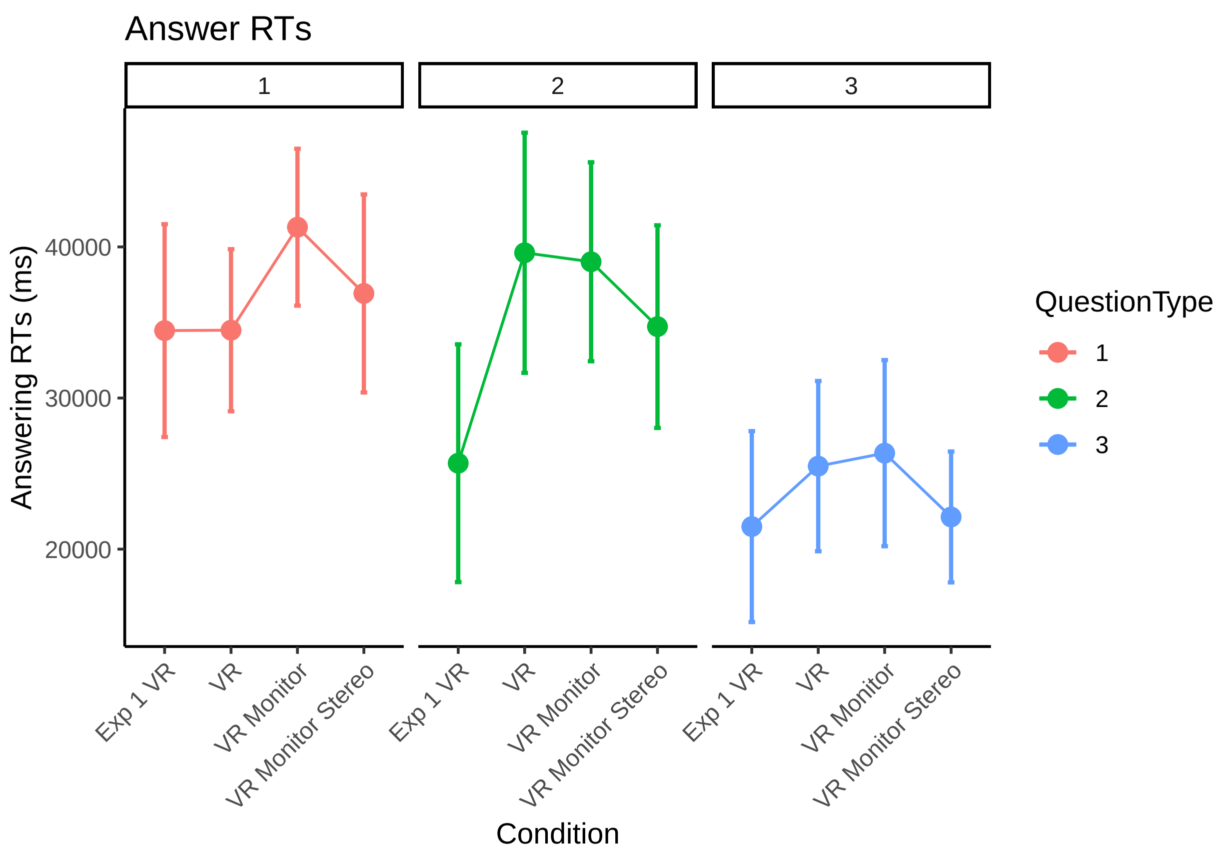


Compare Exp 1 and Exp 2

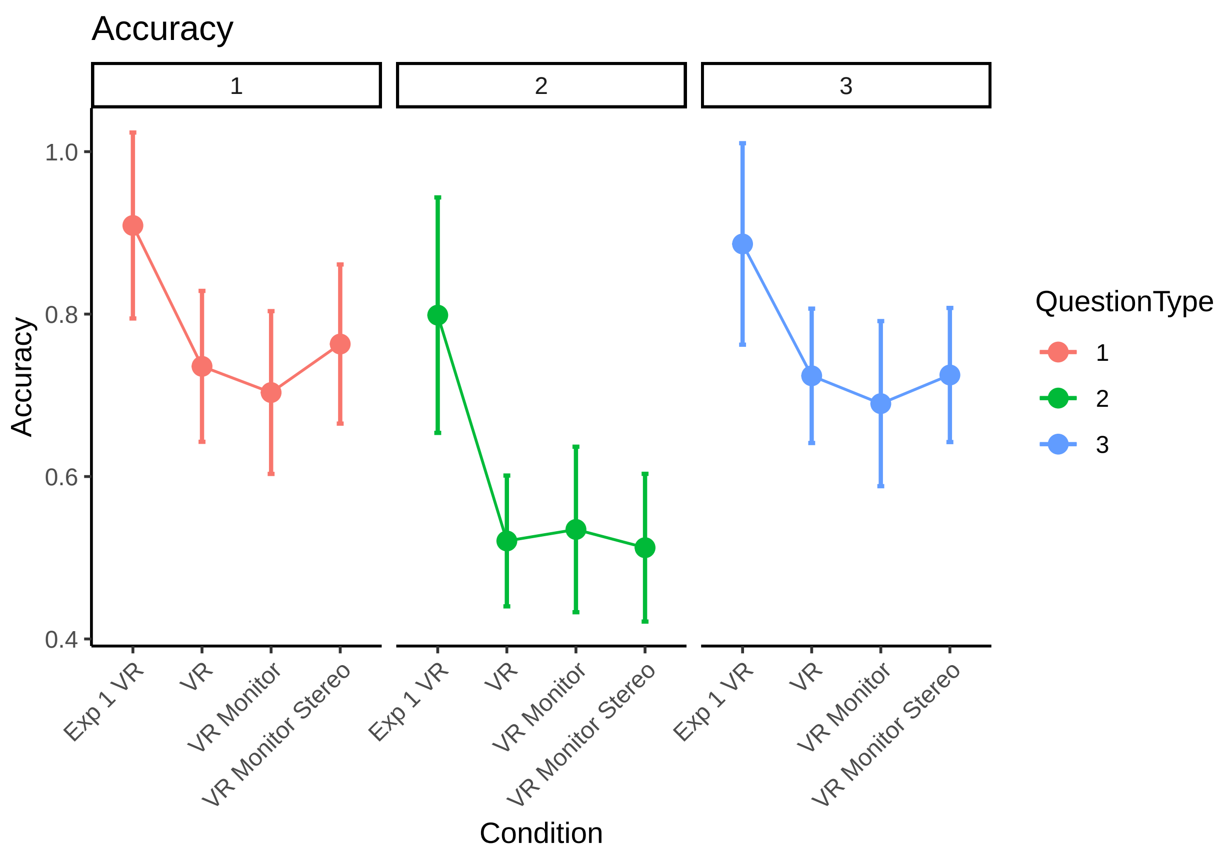
```
exp1_data <- read.spss(file = "Experiment 1/Raw Data.sav",
                       to.data.frame = TRUE) %>%
  select(Participant, M_Condition, ThreeDtype1_acc, ThreeDtype2_acc, ThreeDtype3_acc,
         X.3DTimeInput_IDm, X.3DTimeInput_Relm, X.3DTimeInput_Predm) %>%
  mutate(Condition = str_trim(paste0("Exp 1 ", M_Condition)),
         Participant = as.character(Participant))

exp1_rt_wide <- exp1_data %>%
  dplyr::rename(ParticipantId = Participant,
                Type1 = X.3DTimeInput_IDm,
                Type2 = X.3DTimeInput_Relm,
                Type3 = X.3DTimeInput_Predm) %>%
```

```r
  select(ParticipantId, Condition, Type1, Type2, Type3) %>%
  filter(Condition == "Exp 1 VR")

compare_RT <- bind_rows(answer_rt_type_wider, exp1_rt_wide)

#make the little plot
superbPlot(compare_RT,
    BSFactors    = "Condition",
    WSFactors    = "QuestionType(3)",
    variables    = c("Type1", "Type2", "Type3"),
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose       = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "QuestionType")
) +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
facet_wrap(vars(QuestionType))+
labs(y = "Answering RTs (ms)", title = "Answer RTs")
```

```
## superb::ADVICE: Some of the groups' variances are heterogeneous. Consider using purpose="tryon".
```



```r
exp1_acc_wide <- exp1_data %>%
  dplyr::rename(ParticipantId = Participant,
                Type1 = ThreeDtype1_acc,
                Type2 = ThreeDtype2_acc,
                Type3 = ThreeDtype3_acc) %>%
  select(ParticipantId, Condition, Type1, Type2, Type3) %>%
  filter(Condition == "Exp 1 VR")

compare_acc <- bind_rows(acc_type_wider, exp1_acc_wide)

#make the little plot
superbPlot(compare_acc,
    BSFactors    = "Condition",
    WSFactors    = "QuestionType(3)",
    variables    = c("Type1", "Type2", "Type3"),
    statistic    = "mean",
    errorbar     = "CI",
    gamma        = 0.95,
    adjustments = list(
        purpose       = "difference"
    ),
    plotStyle = "line",
    factorOrder = c("Condition", "QuestionType")
) +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))+
facet_wrap(vars(QuestionType))+
labs(y = "Accuracy", title = "Accuracy")
```




```
# display.acc3D <- data.frame(cbind(data$Participant, data$M_Condition,
#                                   data$ThreeDtype1_acc, data$ThreeDtype2_acc, data$ThreeDtype3_acc))
# display.RT3D <- data.frame(cbind(data$Participant, data$M_Condition,
#                                  data$X.3DTimeInput_IDm, data$X.3DTimeInput_Relm, data$X.3DTimeInput_
```