Visualization Literacy Analysis

Laura Marusich, Jonathan Bakdash

3/22/2022

Read in data files

```
#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("Raw Data")
first_six <- substr(raw_file_names, 1, 6)
sub_ids <- unique(first_six)

length(sub_ids)
```

```
## [1] 122
```

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

rt_data <- NULL

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

  temp_main_file <- read_csv(paste0("Raw Data/", sub_ids[i], "_maindata.csv")) %>%
    mutate(AnswerRT = TimeToBeginInput - TimeToReadQuestion)

  #three potential RTs to exclude by:
  ## total RT (reading + answering)
  ## reading RT
  ## answering RT (i'm thinking this one)

  if (any(temp_main_file$TimeToReadQuestion < 2000, na.rm = T)){
    which_index <- which(temp_main_file$TimeToReadQuestion < 2000)
    for (j in which_index){
      fast_RTs <- add_row(fast_RTs, ParticipantId = sub_ids[i],
                          TrialName = temp_main_file$TrialName[j],
                          type = "ReadingRT",
                          time = temp_main_file$TimeToReadQuestion[j])
    }
  }

  if (any(temp_main_file$AnswerRT < 2000, na.rm = T)){
```

```r
    which_index <- which(temp_main_file$AnswerRT < 2000)
    for (j in which_index){
      fast_RTs <- add_row(fast_RTs, ParticipantId = sub_ids[i],
                          TrialName = temp_main_file$TrialName[j],
                          type = "AnswerRT",
                          time = temp_main_file$AnswerRT[j])
    }
  }

  if (any(temp_main_file$TimeToBeginInput < 2000, na.rm = T)){
    which_index <- which(temp_main_file$TimeToBeginInput < 2000)
    for (j in which_index) {
      fast_RTs <- add_row(fast_RTs, ParticipantId = sub_ids[i],
                          TrialName = temp_main_file$TrialName[j],
                          type = "TotalRT",
                          time = temp_main_file$TimeToBeginInput[j])
    }
  }

  rt_data <- rt_data %>%
    bind_rows(temp_main_file)

}

rt_data <- rt_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)

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

Basic checks

```r
#does everyone have 17 trials

dim(rt_data)[1]
```

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

```r
#122 participants, 17 trials
122*17
```

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

```r
trials_per_participant <- rt_data %>%
  group_by(ParticipantId, Condition) %>%
  summarize(n = n())
```

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

```r
all(trials_per_participant$n == 17)
```

```
## [1] TRUE
```

```r
#how many participants per condition
subs_per_condition <- trials_per_participant %>%
  group_by(Condition) %>%
  summarize(nsubs = n())
#why is the balance so off?
kable(subs_per_condition)
```

Condition	nsubs
VR	50
VR Monitor	39
VR Monitor Stereo	33

Remove outliers

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

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

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

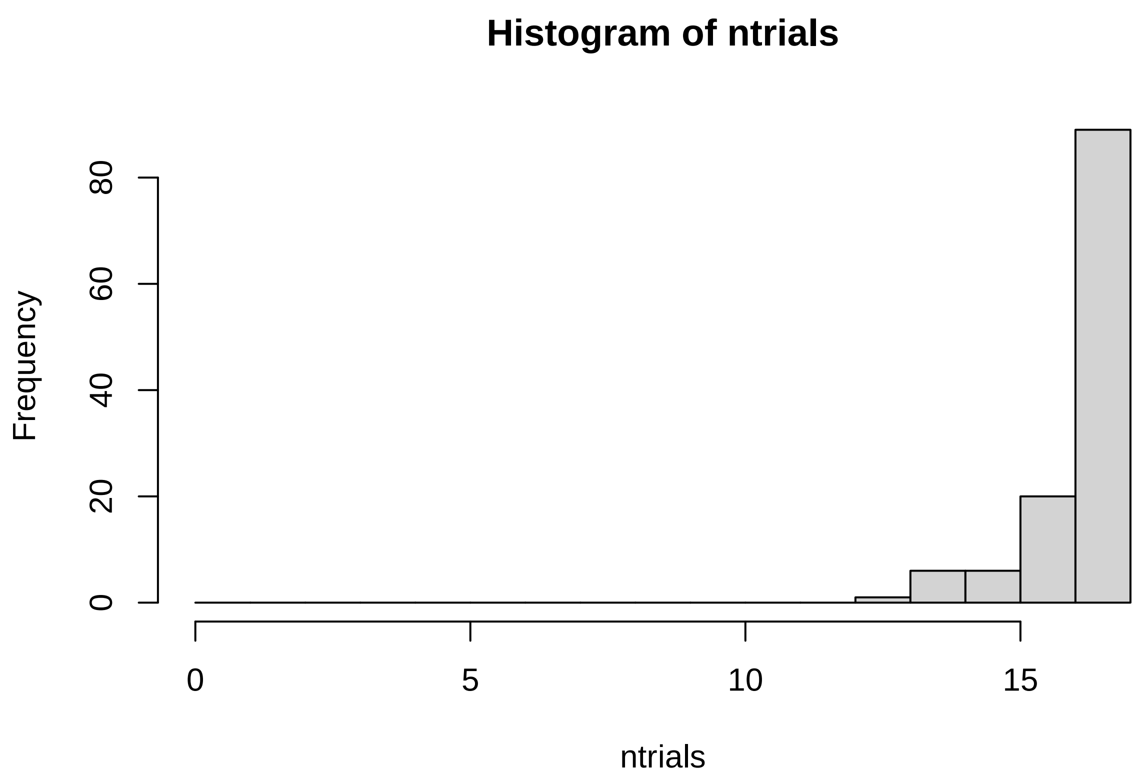
```r
#drops 7 trials

rt_data_summary <- rt_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.
```

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

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

```r
#drops 47 more trials

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




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

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

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


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

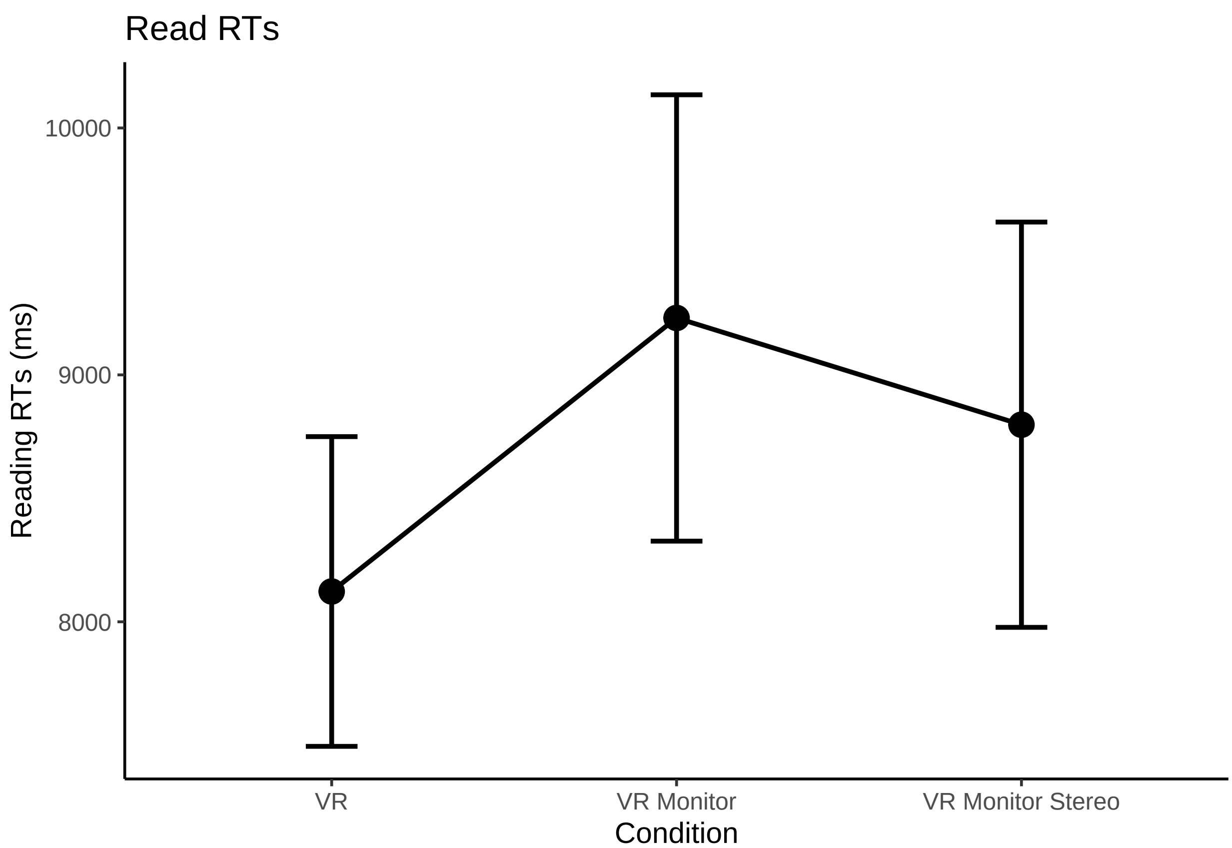
```
            n = n())
```

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

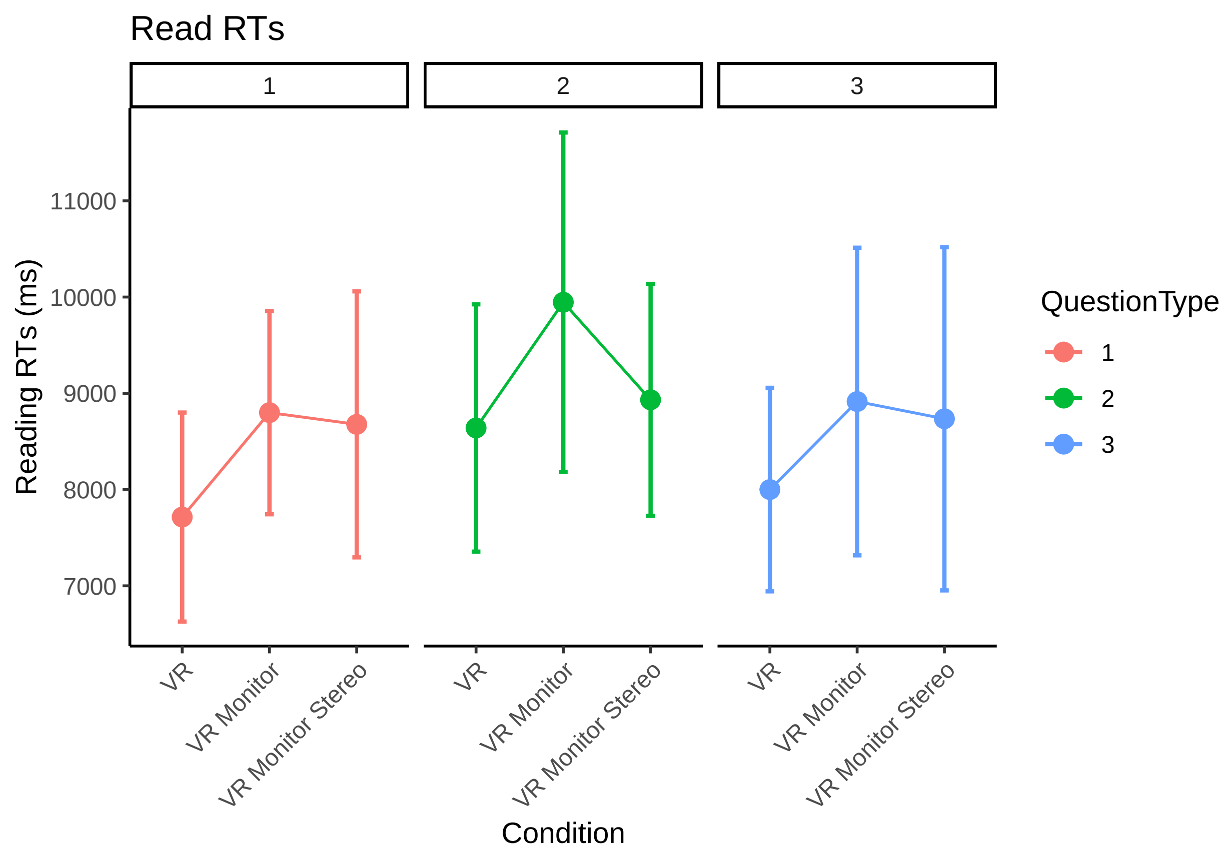
```
# first, make .csv files in wide format to double check in statview
read_rt_type_wider <- trial_type_means %>%
  select(ParticipantId, Condition, TrialType, mean_readRT) %>%
  pivot_wider(names_from = TrialType,values_from = mean_readRT)
answer_rt_type_wider <- trial_type_means %>%
  select(ParticipantId,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


#### READ RTs ####

#make some plots

#just condition main effect
readplot1 <- rt_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.
```

```
 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")
```




```r
#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")
```




```r
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
```

```r
read_rt_type_anova
```

```
## Anova Table (Type 3 tests)
##
## Response: mean_readRT
##                Effect          df         MSE      F  pes p.value
## 1           Condition      2, 119 18051925.11   2.28 .037    .107
## 2           TrialType 2.00, 237.70  4660052.21 4.30 * .035    .015
## 3 Condition:TrialType 4.00, 237.70  4660052.21   0.52 .009    .723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
##
## Sphericity correction method: GG
```

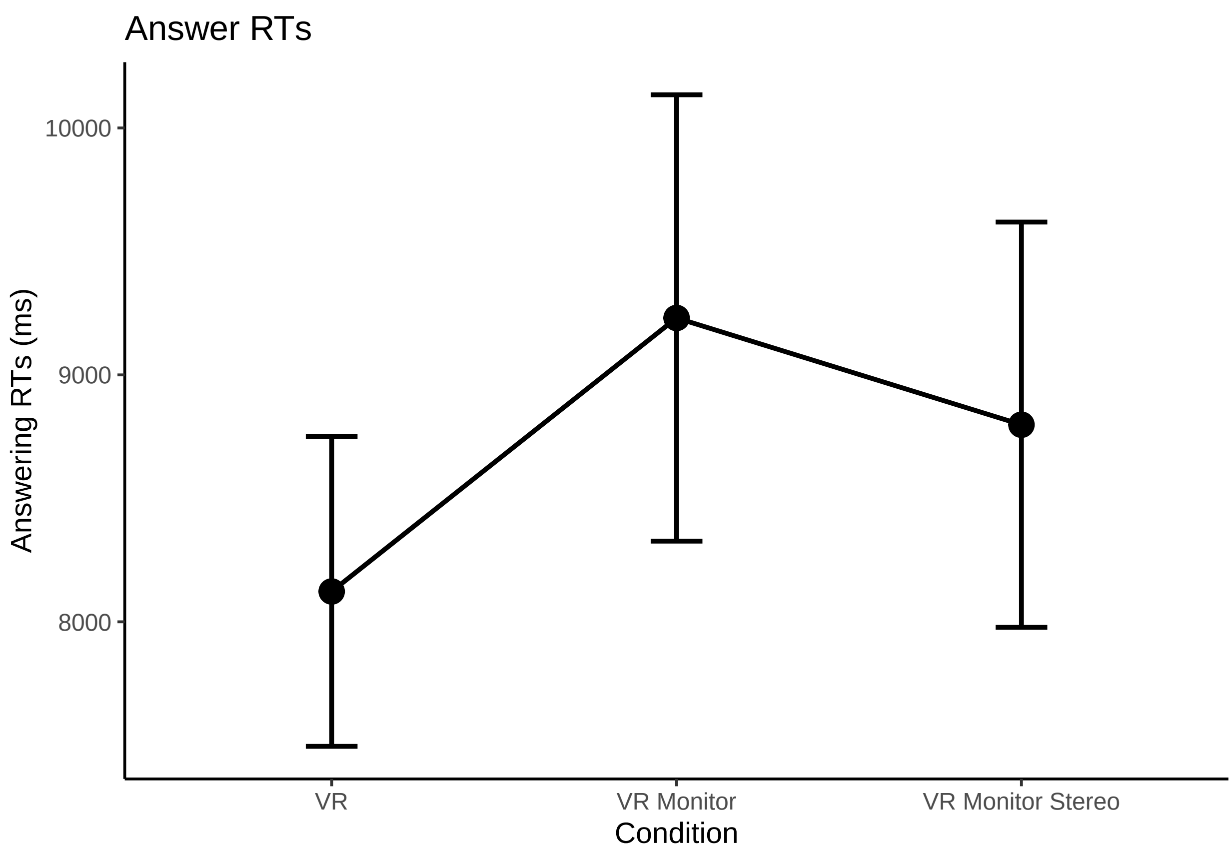
```r
#posthoc test for trial type
pairs(emmeans(read_rt_type_anova, "TrialType"), adjust = "Tukey")
```

```
##  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
```

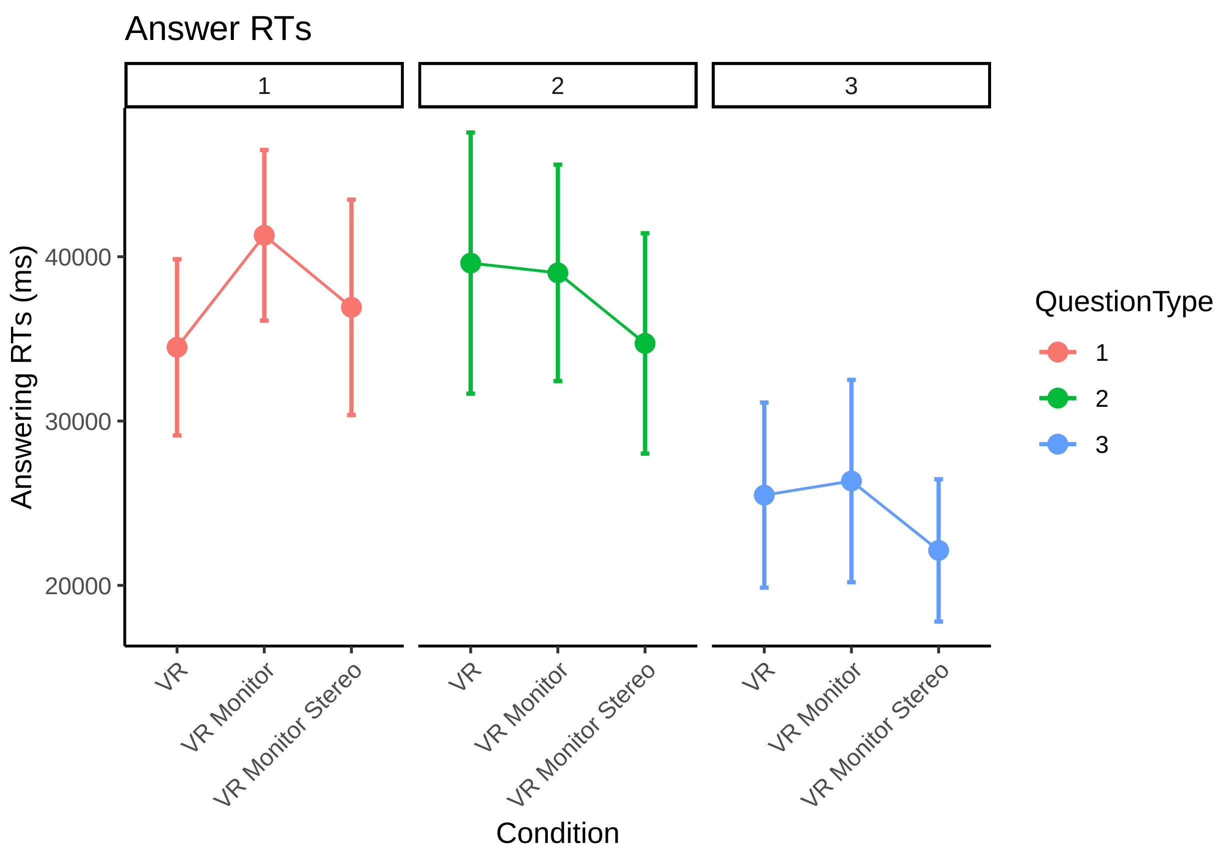
```
#Question Type 2 slower than Type 1, marginally slower than Type 3 (for reading times)


#### ANSWER RTs ####

#make some plots

#just condition main effect
answerplot1 <- rt_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(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 = "Answering RTs (ms)", title = "Answer RTs")
```




```r
#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
answer_rt_type_anova
```

```
## Anova Table (Type 3 tests)
##
## Response: mean_answerRT
##                Effect          df          MSE         F  pes p.value
## 1           Condition       2, 119 415942756.23      1.21 .020    .303
## 2           TrialType 1.93, 229.53  92112713.90 75.40 *** .388   <.001
## 3 Condition:TrialType 3.86, 229.53  92112713.90    2.52 * .041    .044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
##
## Sphericity correction method: GG
```

```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
```