

Visualization Literacy Analysis

Laura Marusich

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$AnswerRT < 2000, na.rm = T)){
  which_index <- which(temp_main_file$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 = as.factor(TrialType))

```

Remove outliers

```
dim(rt_data)[1]
```

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

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

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

```
#remove answerRTs below 2000ms first
rt_data_remove <- rt_data %>%

```

```
filter(AnswerRT >= 2000)
dim(rt_data_remove)[1]
```

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

#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.
```

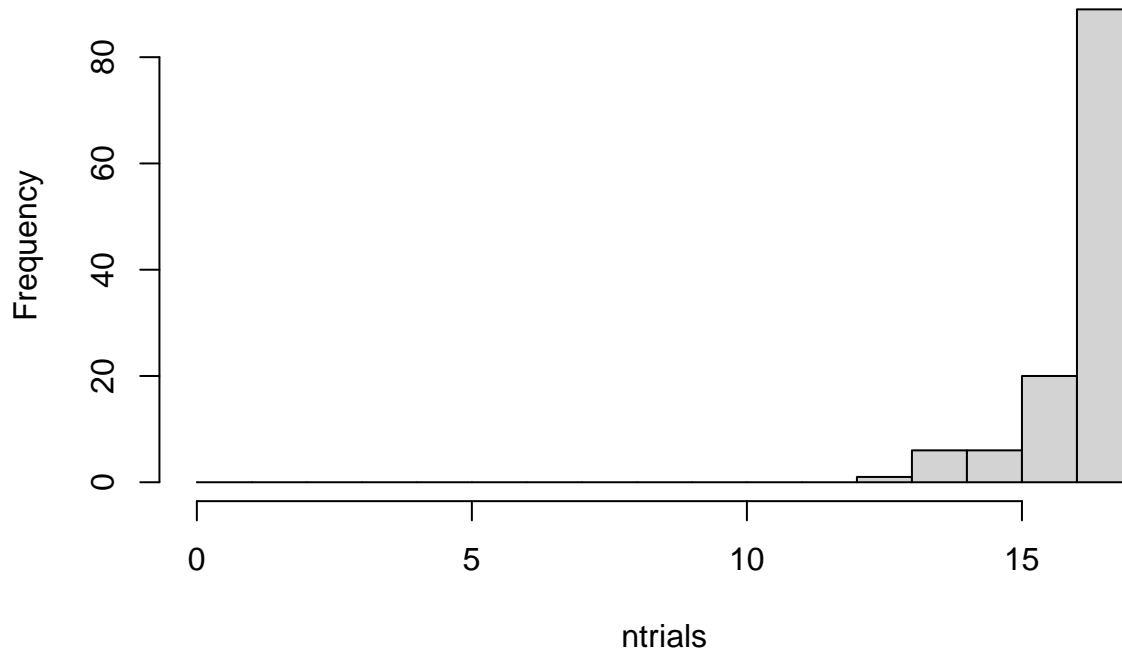
```
rt_data_no_outliers <- rt_data_remove %>%
  group_by(TrialName) %>%
  filter(!(abs(AnswerRT - mean(AnswerRT)) > 3*sd(AnswerRT)))
dim(rt_data_no_outliers)[1]
```

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

#drops 47 more trials

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

Histogram of ntrials



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

Compare conditions using individual questions

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

# first, make .csv files in wide format to double check in statview
read_rt_wider <- rt_data_no_outliers %>%
  select(ParticipantId, Condition, TrialName, readRT) %>%
  pivot_wider(names_from = TrialName, values_from = readRT) %>%
  mutate(noutliers = rowSums(is.na(across(3:19))))

answer_rt_wider <- rt_data_no_outliers %>%
  select(ParticipantId, Condition, TrialName, AnswerRT) %>%
  pivot_wider(names_from = TrialName, values_from = AnswerRT) %>%
  mutate(noutliers = rowSums(is.na(across(3:19))),
         mean_all = rowMeans(across(3:19), na.rm = T))

write.csv(read_rt_wider, file = "readRTs.csv", row.names = F)
write.csv(answer_rt_wider, file = "answerRTs.csv", row.names = F)

#within ANOVA can't run with missing cells, so remove participants that had outliers removed
```

```

#this loses a lot of data. would probably be better to replace outliers with means, but
#i figure we don't really want to use individual question as the within variable anyway
#see below for analysis with question TYPE as the within variable
rt_data_no_outliers2 <- rt_data_no_outliers %>%
  group_by(ParticipantId) %>%
  filter(n() > 16)

```

#READING RTs

#make a little plot

```

readplot1 <- rt_data_no_outliers2 %>%
  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)

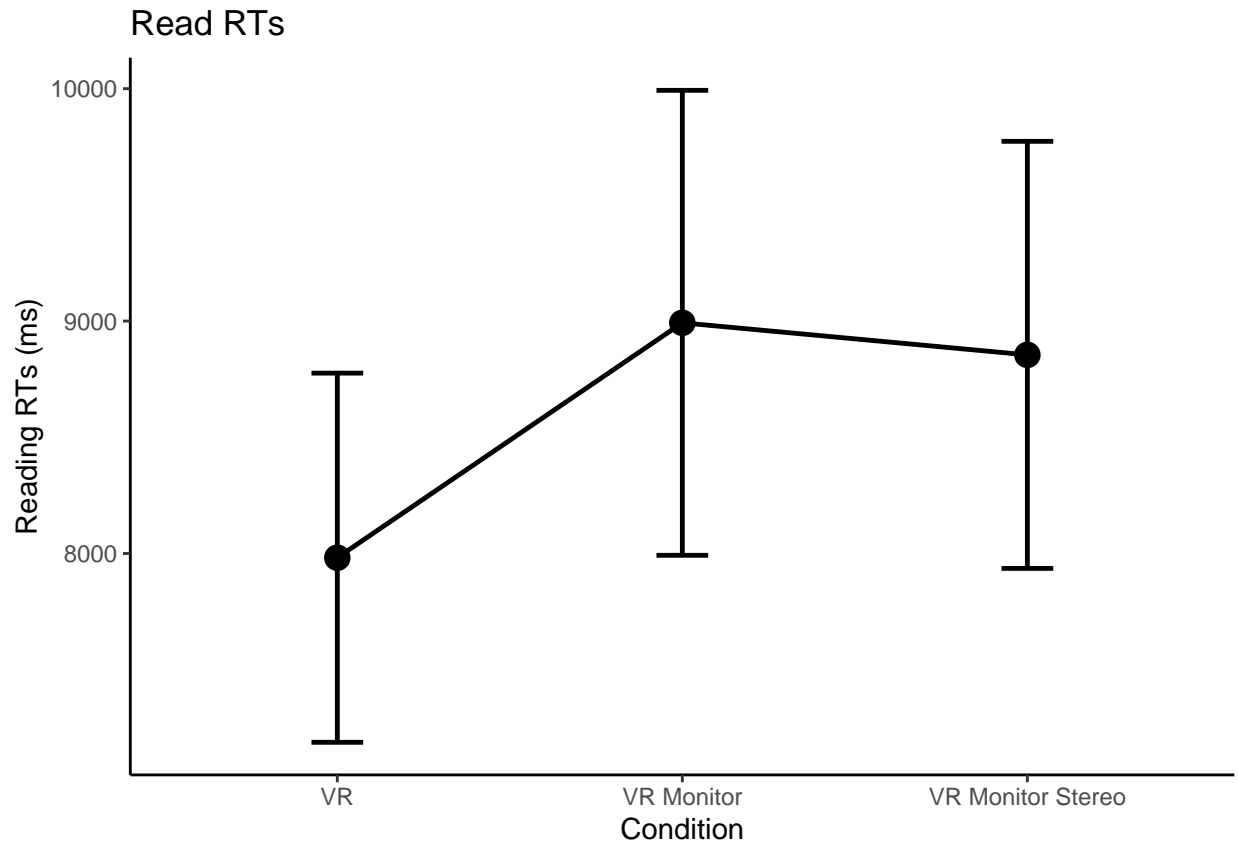
```

'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")

```



```
read_rt_anova <- ezANOVA(rt_data_no_outliers2,
  dv = readRT,
  wid = ParticipantId,
  within = TrialName,
  between = Condition,
  type = 3,
  detailed = TRUE,
  return_aov = T
)
```

```
## Warning: Converting "ParticipantId" to factor for ANOVA.
```

```
## Warning: Converting "TrialName" to factor for ANOVA.
```

```
## Warning: Converting "Condition" to factor for ANOVA.
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-
## considered value for the type argument to ezANOVA().
```

```
#get the partial eta-squared too
aovEffectSize(read_rt_anova, effectSize = "pes")$ANOVA
```

```
##          Effect DFn  DFd      SSn      SSd      F      p p<.05
## 1 (Intercept)    1   86 110050120600 8403185411 1126.276514 3.448191e-51 *
```

```
## 2          Condition  2   86   327980221  8403185411   1.678310 1.927454e-01
## 3          TrialName 16 1376 13109978306 38691894000 29.139389 6.716381e-76      *
## 4 Condition:TrialName 32 1376   944710922 38691894000   1.049899 3.919419e-01
##          pes
## 1 0.92905909
## 2 0.03756431
## 3 0.25307924
## 4 0.02383430
```

```
#no significant effect of condition on read times
```

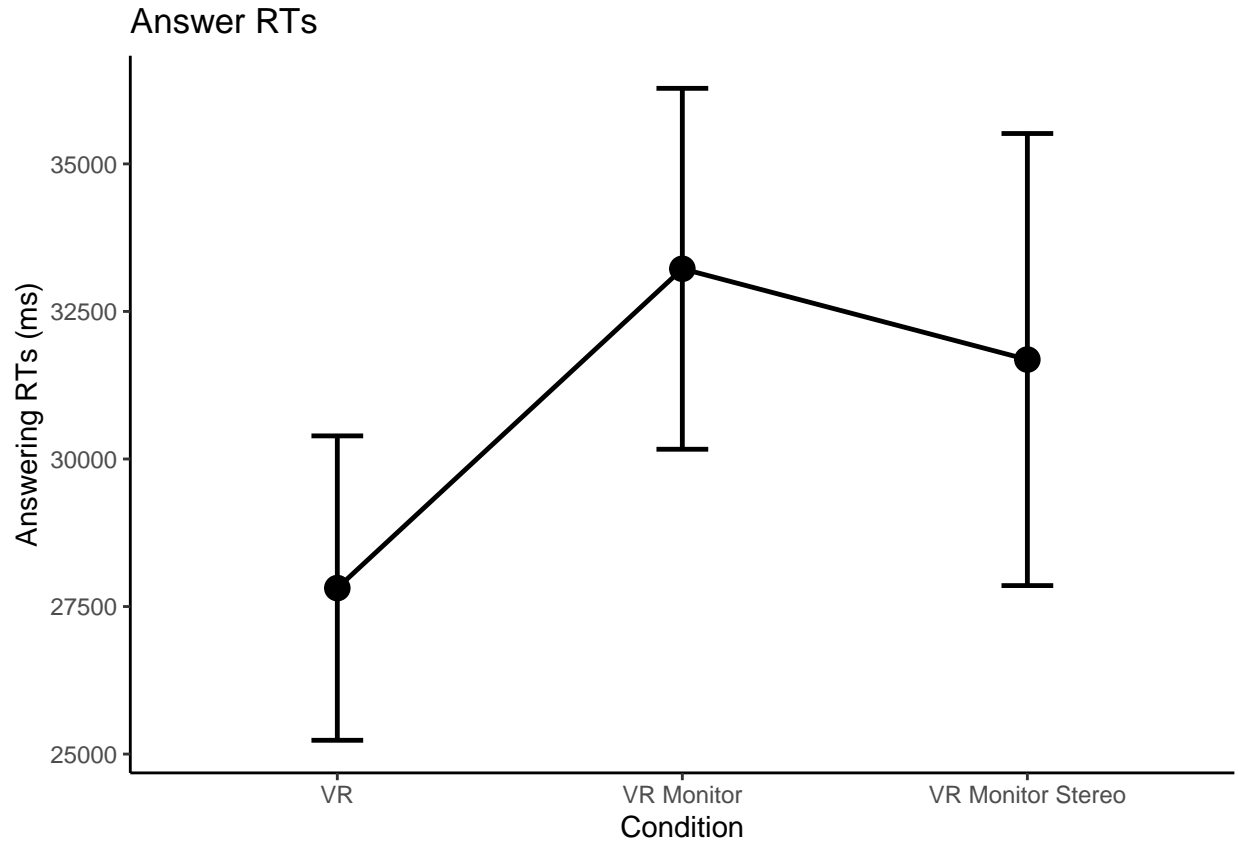
```
#ANSWERING RTs
```

```
#make a little plot
```

```
answerplot1 <- rt_data_no_outliers2 %>%
  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")
```



```
answer_rt_anova <- ezANOVA(rt_data_no_outliers2,
  dv = AnswerRT,
  wid = ParticipantId,
  within = TrialName,
  between = Condition,
  type = 3,
  detailed = TRUE,
  return_aov = T
)
```

```
## Warning: Converting "ParticipantId" to factor for ANOVA.
```

```
## Warning: Converting "TrialName" to factor for ANOVA.
```

```
## Warning: Converting "Condition" to factor for ANOVA.
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-
## considered value for the type argument to ezANOVA().
```

```
aovEffectSize(answer_rt_anova, effectSize = "pes")$ANOVA
```

##	Effect	DFn	DFd	SSn	SSd	F	p	p<.05
## 1	(Intercept)	1	86	1.418165e+12	9.84632e+10	1238.658028	7.596851e-53	*
## 2	Condition	2	86	8.509034e+09	9.84632e+10	3.715992	2.832199e-02	*


```
## 3          TrialName 16 1376 3.160056e+11 5.13627e+11 52.910937 9.625859e-131 *
## 4 Condition:TrialName 32 1376 1.329243e+10 5.13627e+11 1.112821 3.051122e-01
##          pes
## 1 0.93507758
## 2 0.07954433
## 3 0.38089828
## 4 0.02522669
```

#Post-hoc tests

#Remove Within Factor of TrialName (17 repeated question levels), causes problems with emmeans?

```
answer_rt_anova2 <- ezANOVA(rt_data_no_outliers2,
                             dv = AnswerRT,
                             wid = ParticipantId,
                             #within = TrialName,
                             between = Condition,
                             type = 3,
                             detailed = F,
                             return_aov = T
                           )
```

```
## Warning: Converting "ParticipantId" to factor for ANOVA.
```

```
## Warning: Converting "Condition" to factor for ANOVA.
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-
## considered value for the type argument to ezANOVA().
```

```
## Warning: Collapsing data to cell means. *IF* the requested effects are a subset of the
## full design, you must use the "within_full" argument, else results may be inaccurate.
```

```
## Coefficient covariances computed by hccm()
```

```
emm.RT.answer <- emmeans(answer_rt_anova2$aov, ~ Condition)
emm.RT.answer
```

```
## Condition      emmean    SE df lower.CL upper.CL
## VR              27812 1387 86    25055    30570
## VR Monitor      33223 1524 86    30193    36252
## VR Monitor Stereo 31685 1641 86    28422    34948
##
## Confidence level used: 0.95
```

```
pairs(emm.RT.answer, adjust = "Tukey")
```

```
## contrast              estimate    SE df t.ratio p.value
## VR - VR Monitor          -5410 2061 86   -2.625  0.0274
## VR - VR Monitor Stereo   -3872 2149 86   -1.802  0.1750
## VR Monitor - VR Monitor Stereo 1538 2240 86    0.687  0.7719
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Compare conditions for question type (instead of by individual question)

```
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 '.groups' argument.

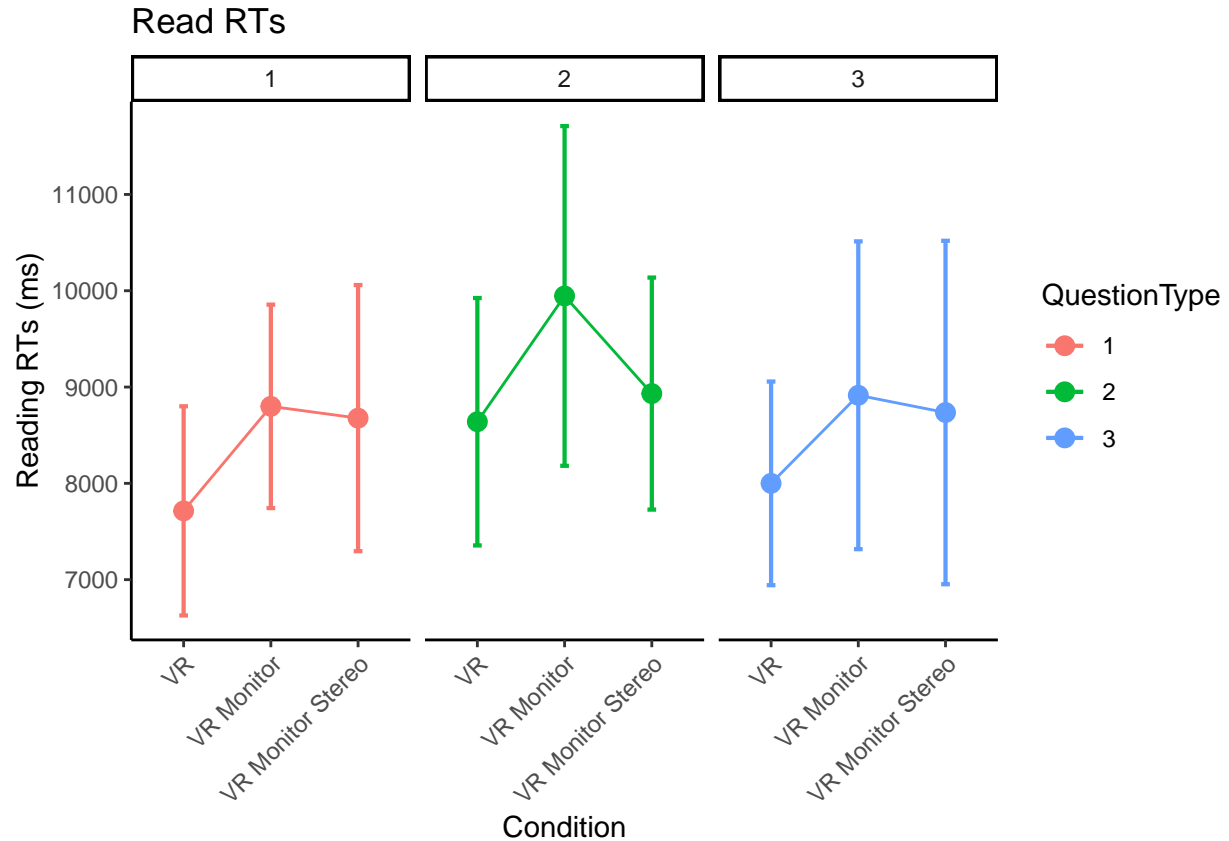
```
read_rt_type_wider <- trial_type_means %>%
  select(ParticipantId, Condition, TrialType, mean_readRT) %>%
  pivot_wider(names_from = TrialType, values_from = mean_readRT, names_prefix="Type")
answer_rt_type_wider <- trial_type_means %>%
  select(ParticipantId, Condition, TrialType, mean_answerRT) %>%
  pivot_wider(names_from = TrialType, values_from = mean_answerRT, names_prefix="Type")
```

```
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 the little plot

```
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 <- ezANOVA(trial_type_means,
                              dv = mean_readRT,
                              wid = ParticipantId,
                              within = TrialType,
                              between = Condition,
                              type = 3,
                              detailed = TRUE,
                              return_aov = T
)
```

```
## Warning: Converting "ParticipantId" to factor for ANOVA.
```

```
## Warning: Converting "Condition" to factor for ANOVA.
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-
## considered value for the type argument to ezANOVA().
```

```
#get the partial eta-squared too
aovEffectSize(read_rt_type_anova, effectSize = "pes")$ANOVA
```

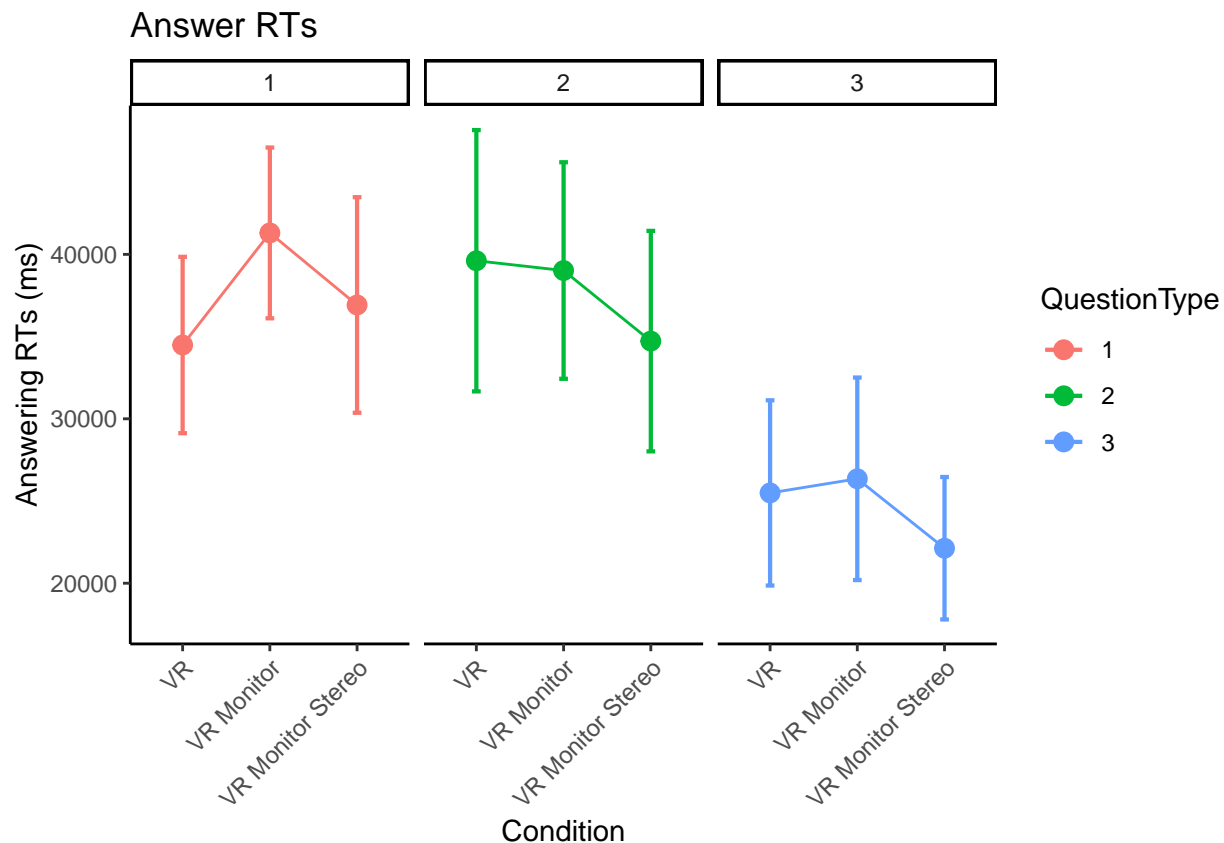
##	Effect	DFn	DFd	SSn	SSd	F	p	p<.05
## 1	(Intercept)	1	119	26950178706	2148179088	1492.9254659	3.448588e-69	*
## 2	Condition	2	119	82182691	2148179088	2.2762861	1.071180e-01	
## 3	TrialType	2	238	40014704	1107710827	4.2987300	1.465597e-02	*

```
## 4 Condition:TrialType    4 238      9637742 1107710827    0.5176853 7.228079e-01
##           pes
## 1 0.926175247
## 2 0.036847247
## 3 0.034864349
## 4 0.008625546
```

```
#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")
```



```
answer_rt_type_anova <- ezANOVA(trial_type_means,
                                dv = mean_answerRT,
                                wid = ParticipantId,
                                within = TrialType,
                                between = Condition,
                                type = 3,
                                detailed = TRUE,
                                return_aov = T
)
```

```
## Warning: Converting "ParticipantId" to factor for ANOVA.
```

```
## Warning: Converting "Condition" to factor for ANOVA.
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-
## considered value for the type argument to ezANOVA().
```

```
#get the partial eta-squared too
aovEffectSize(answer_rt_type_anova, effectSize = "pes")$ANOVA
```

##	Effect	DFn	DFd	SSn	SSd	F	p	p<.05
## 1	(Intercept)	1	119	395128217295	49497187991	949.958165	1.446491e-58	*
## 2	Condition	2	119	1004601416	49497187991	1.207620	3.025441e-01	
## 3	TrialType	2	238	13395765201	21142570734	75.397457	4.324238e-26	*

```
## 4 Condition:TrialType    4 238    896611493 21142570734    2.523269 4.168886e-02    *
##          pes
## 1 0.88867665
## 2 0.01989239
## 3 0.38785207
## 4 0.04068261
```