

Visualization Literacy Analysis

Laura Marusich, Jonathan Bakdash

3/22/2022

Read in data files

```
#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)
```

```

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

```

#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

```

#why is the balance so off?

```

Remove outliers based on RT

```

#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
```

```
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.
```

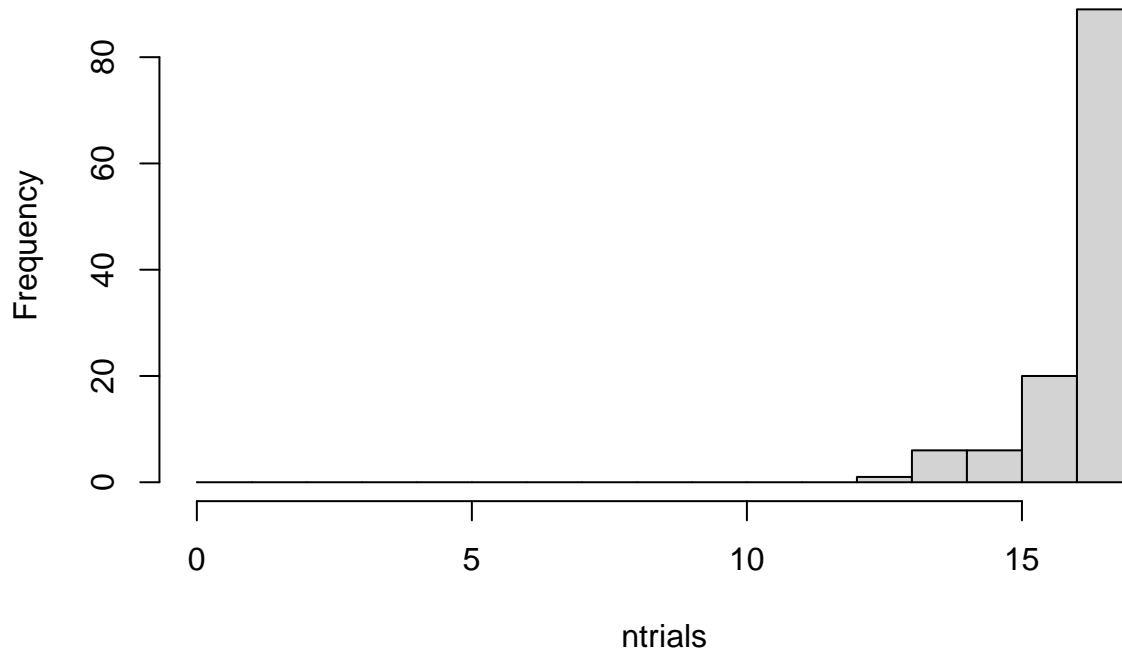
```
all_data_no_outliers <- all_data_remove %>%  
  group_by(TrialName) %>%  
  filter(!(abs(AnswerRT - mean(AnswerRT)) > 3*sd(AnswerRT)))  
dim(all_data_no_outliers)[1]
```

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

```
#drops 47 more trials
```

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

Histogram of ntrials



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

```
#### READ RTs ####
```

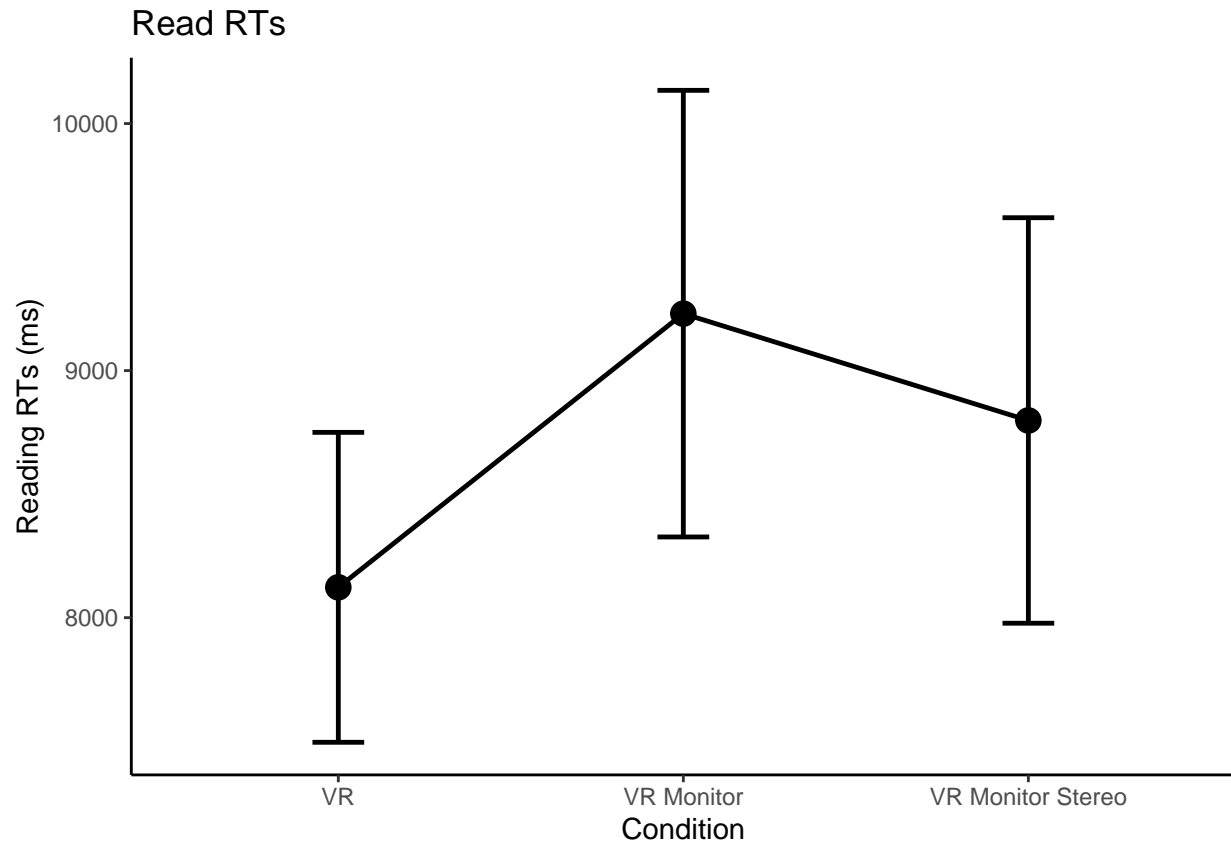
```
#make some plots
```

```
#just condition main effect
```

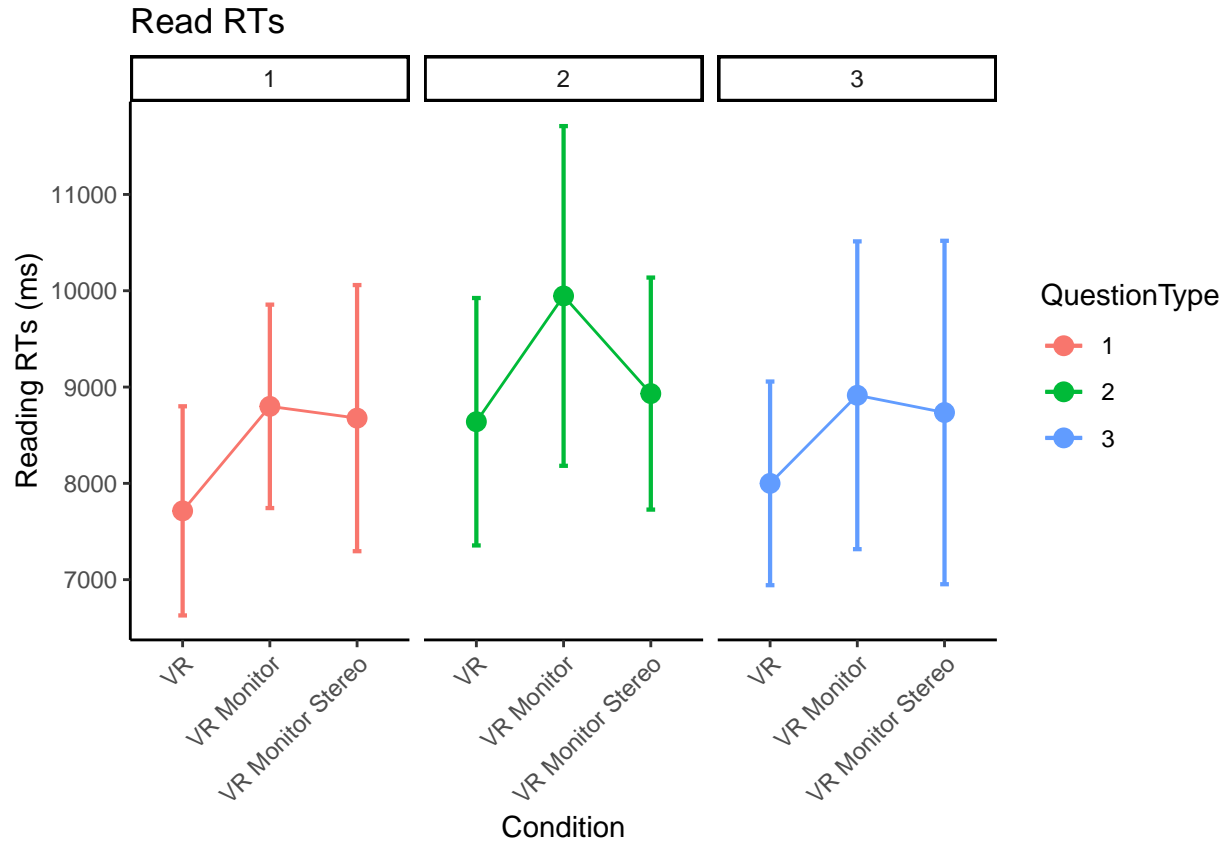
```
readplot1 <- all_data_no_outliers %>%  
  group_by(ParticipantId, Condition) %>%  
  summarize(overallmean = mean(readRT)) %>%  
  group_by(Condition) %>%  
  summarize(overall_condition_mean = mean(overallmean),  
            se = std.error(overallmean),  
            n = n(),  
            CI =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

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

```
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
```

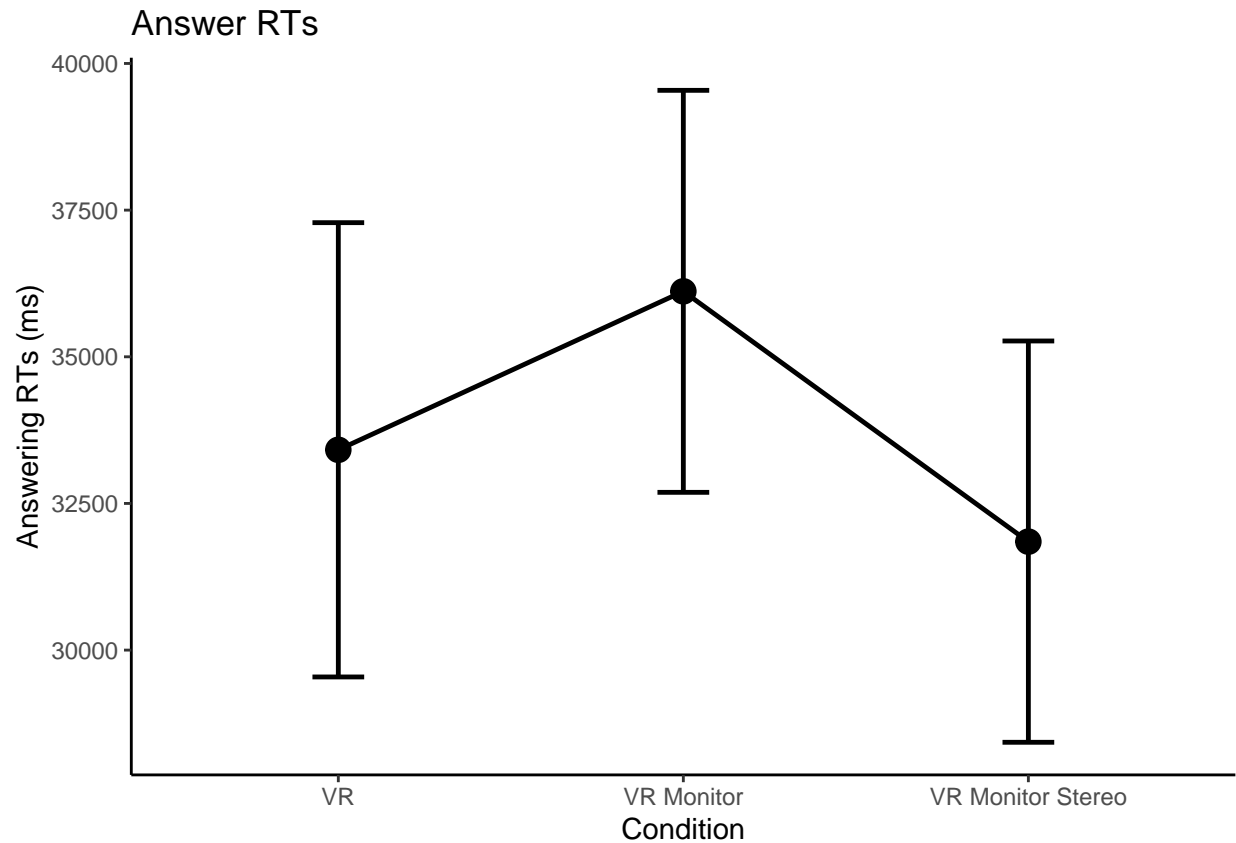
```
##
## Results are averaged over the levels of: Condition
## P value adjustment: tukey method for comparing a family of 3 estimates
#Question Type 2 slower than Type 1, marginally slower than Type 3 (for reading times)

#### ANSWER RTs ####

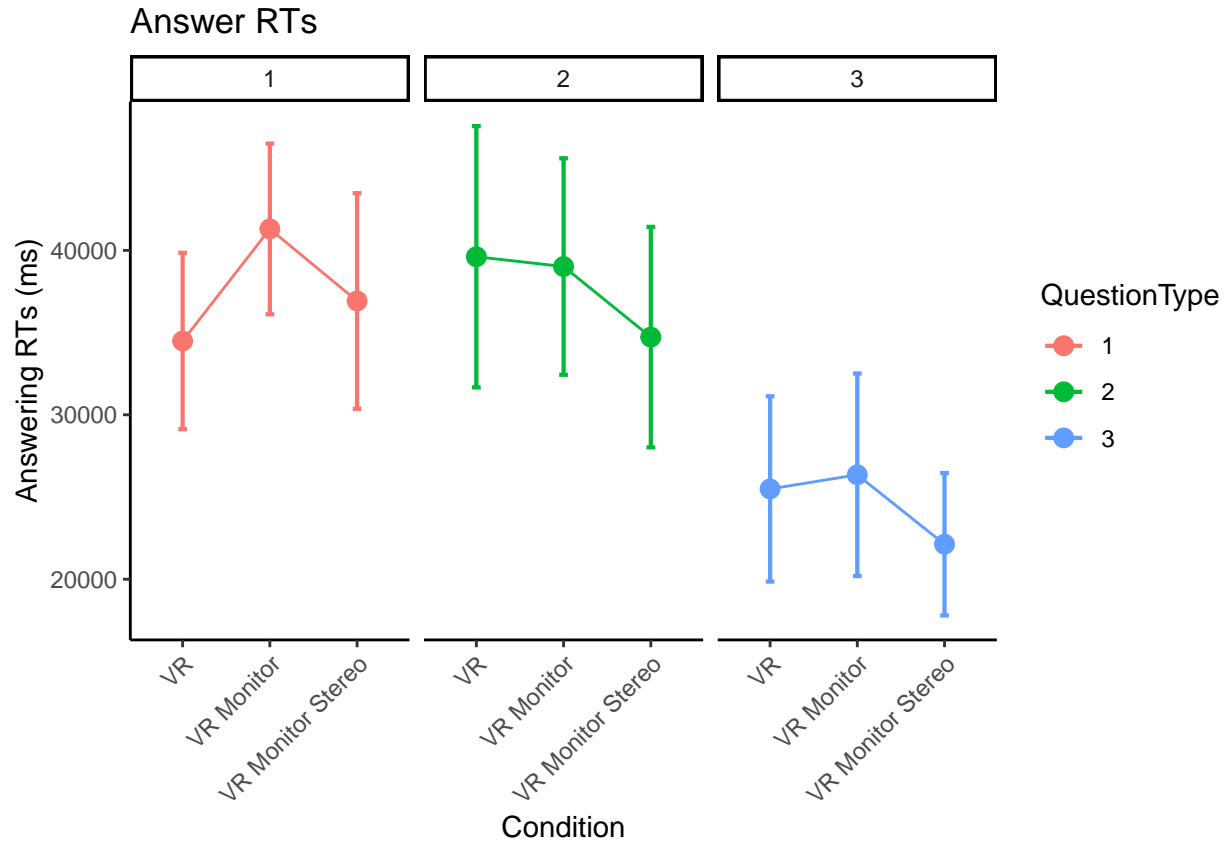
#make some plots

#just condition main effect
answerplot1 <- 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")
```



```
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
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

```
#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
#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
#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
```

Accuracy

```
# do interrater reliability?
# account for non-independence? idk
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

all_data_no_outliers <- all_data_no_outliers %>%
  rowwise() %>%
```

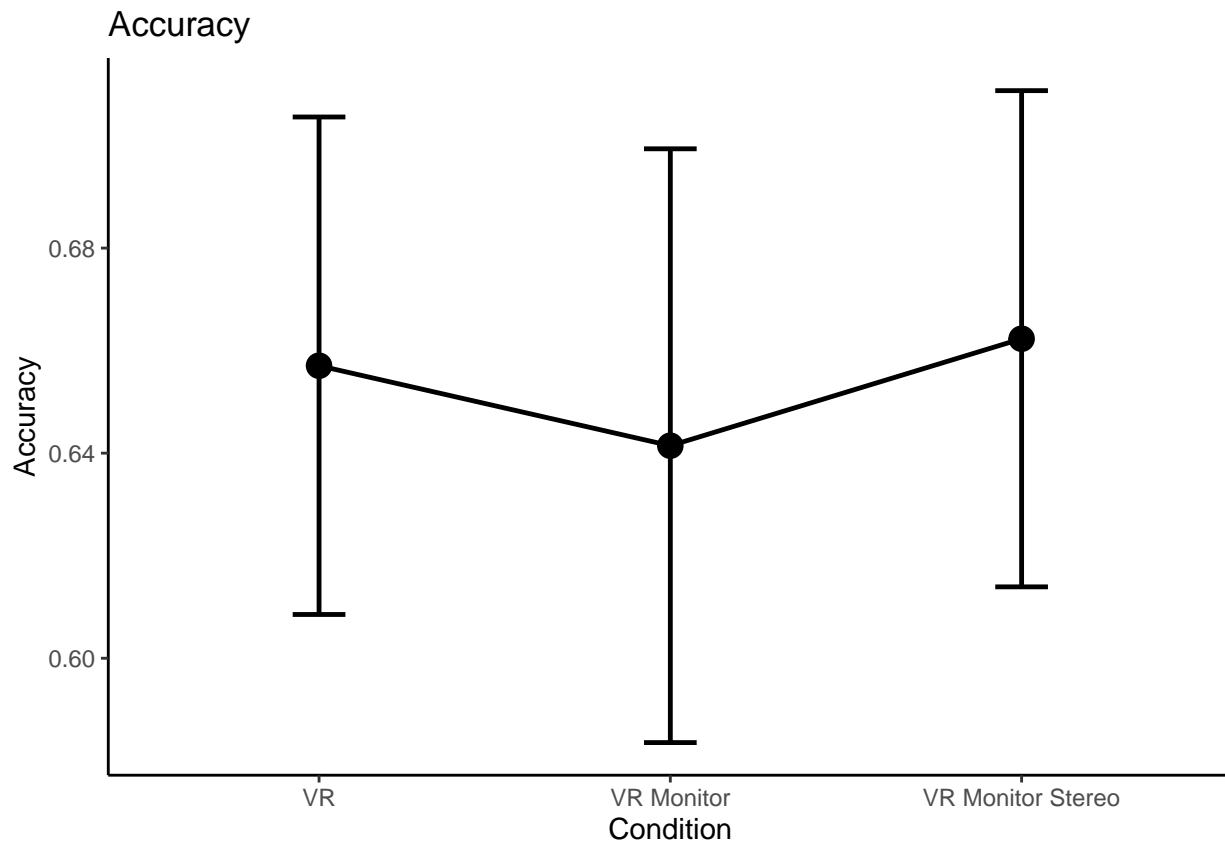
```

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.
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) %>%

```

```

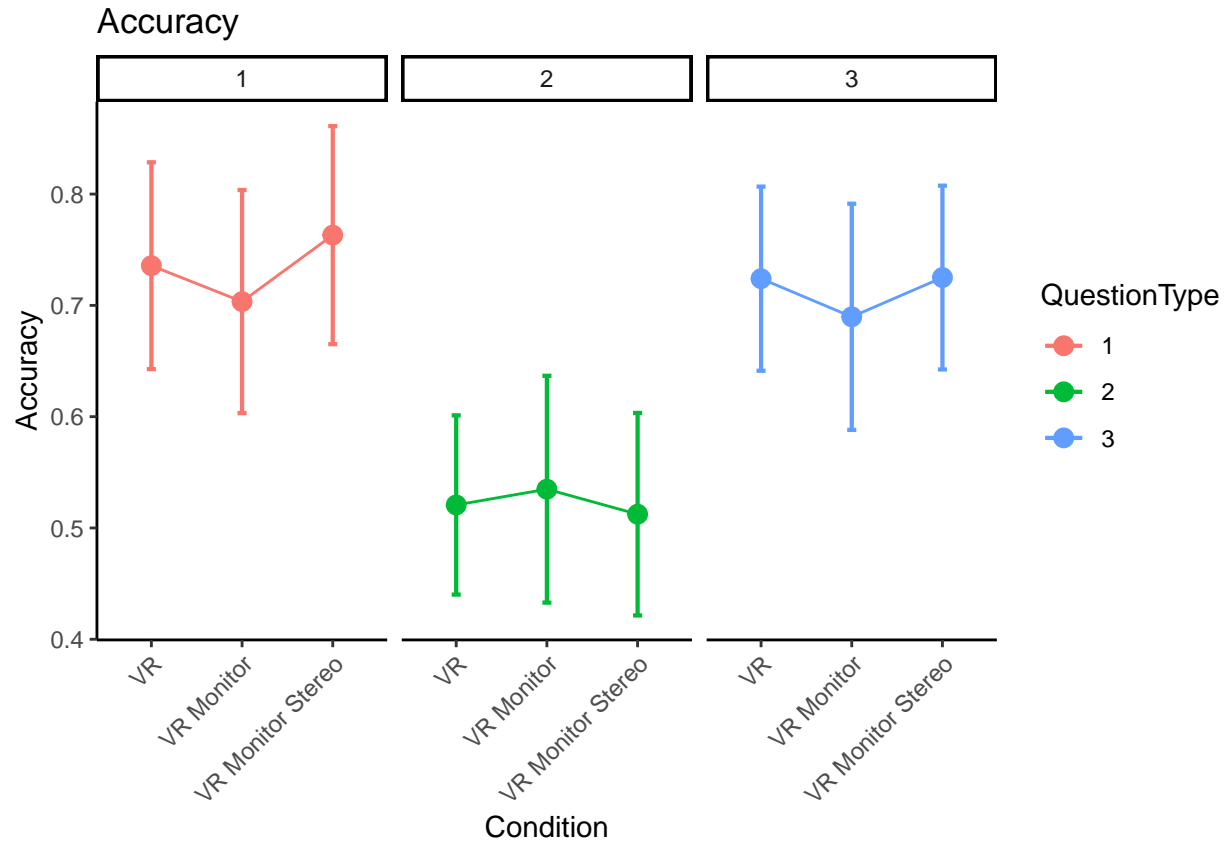
summarize(mean_acc = mean(Correct_Avg),
          n = n())

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

```



```
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
```

```
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