flowers-pilot-analysis

Pre-process

Remove joined_data and uncomment the following block to rerun the preprocessing scripts

```
# data_list <- c("technical_pilots", "collabmulti_pilot0",
#                "collabmulti_20", "compet_20", "multi_nonlang_20")
# process_all_data(data_list)
# join_all_data()
```

```
d.players <- read_csv(here("data/processed_data/joined_data/players.csv"),
                      col_types = cols()) %>% distinct()
d.games <- read_csv(here("data/processed_data/joined_data/games.csv"),
                      col_types = cols()) %>% distinct()
d.rounds <- read_csv(here("data/processed_data/joined_data/rounds.csv"),
                      col_types = cols()) %>% distinct()
d.raw_chat <- read_csv(here("data/processed_data/joined_data/raw_chat.csv"),
                      col_types = cols()) %>% distinct()
d.contexts <- read_csv(here("data/processed_data/joined_data/contexts.csv"),
                      col_types = cols()) %>% distinct()
d.feedback <- read_csv(here("data/processed_data/joined_data/feedback.csv"),
                      col_types = cols()) %>% distinct()
d.demographics <- read_csv(here("data/processed_data/joined_data/demographics.csv"),
                      col_types = cols()) %>% distinct()
```

```
#exclude BLOCKS with <12 completed (all 3 participants responded) rounds
#exclude trial runs

rounds_exclude <- d.rounds %>% group_by(gameId, blockNum) %>% tally() %>% filter(n!=36) %>% select(game

d.rounds.final <-  d.rounds %>% anti_join(rounds_exclude)
```

```
## Joining, by = c("gameId", "blockNum")
```

```
d.raw_chat.final <- d.raw_chat %>% anti_join(rounds_exclude) %>% write_csv(here("data/processed_data/joi
```

```
## Joining, by = c("gameId", "blockNum")
```

```
game_completion <- d.rounds %>%
  group_by(gameId) %>%
  summarize(num_rounds=max(trialNum+1)) %>%
  left_join(d.games%>% ungroup() %>% select(gameId, condition, chat=chatEnabled, gameLength)) %>%
  mutate(gameComplete = ifelse(num_rounds == 48, TRUE, FALSE))
```

```
## Joining, by = "gameId"
```

```
#knitr::kable(summary)
message("Full games")
```

```
## Full games
```

```
game_completion %>% filter(num_rounds==48)
```

```
## # A tibble: 80 x 6
##    gameId             num_rounds condition    chat  gameLength gameComplete
##    <chr>                   <dbl> <chr>        <lgl>      <dbl> <lgl>
##  1 2yX3k7DkPSWGee3q9          48 competCartel TRUE        2610 TRUE
##  2 39CAZTKLsCKbbfEgn          48 coopMulti    FALSE        706 TRUE
##  3 3LgkjpqtmuKooAMZM          48 coopMulti    FALSE        434 TRUE
##  4 4iqSW6pdiYRqJZZBD          48 coopMulti    TRUE         551 TRUE
##  5 5j9HePvQJgnrHefDq          48 coopMulti    TRUE        1611 TRUE
##  6 5oqcuFaDgbZ4pAnvK          48 competCartel TRUE         613 TRUE
##  7 66vT9mE6yqptQDdXG          48 coopCartel   TRUE        1531 TRUE
##  8 67qxzPDuhEbrkhDLF          48 competCartel TRUE        2837 TRUE
##  9 6cWN32G7kFqz6XLFe          48 competCartel TRUE         550 TRUE
## 10 6o4WJuZBT3QSNRQRq          48 coopCartel   TRUE        1978 TRUE
## # ... with 70 more rows
```

```
message("Partial games")
```

```
## Partial games
```

```
game_completion %>% filter(num_rounds!=48)
```

```
## # A tibble: 10 x 6
##    gameId             num_rounds condition    chat  gameLength gameComplete
##    <chr>                   <dbl> <chr>        <lgl>      <dbl> <lgl>
##  1 aF49vDgYfxSWj9bPc          43 coopMulti    TRUE        2092 FALSE
##  2 bJL8HJjyykG9B4k9C           7 competCartel TRUE        4341 FALSE
##  3 Ex5iSXjrMTp3S6zqY           8 coopMulti    FALSE        272 FALSE
##  4 ExXQbCn35Xznf9tM5          36 competCartel TRUE        4922 FALSE
##  5 K3BiGhNQsyeoGmKhr           4 competCartel TRUE         322 FALSE
##  6 MXgWvTXxhRYZka4fo           7 coopMulti    FALSE        444 FALSE
##  7 vomoob4XzDkrw5rXZ           7 competCartel TRUE         674 FALSE
##  8 xgdZ83eD2XxDzFXg7           3 coopMulti    FALSE        219 FALSE
##  9 YmfiWLZCjuweB8ubM           8 coopMulti    FALSE        269 FALSE
## 10 ZRbnTDRALnkiCv2fD          15 competCartel TRUE         586 FALSE
```

Game counts

```
#pull out pilots
d.games %>% group_by(condition, chatEnabled,pilot)%>% tally()
```

```
## # A tibble: 8 x 4
## # Groups:   condition, chatEnabled [6]
##   condition    chatEnabled pilot     n
##   <chr>        <lgl>       <lgl> <int>
## 1 competCartel FALSE       TRUE      5
## 2 competCartel TRUE        FALSE    20
## 3 competCartel TRUE        TRUE      7
## 4 coopCartel   FALSE       TRUE      5
## 5 coopCartel   TRUE        TRUE     10
## 6 coopMulti    FALSE       FALSE    20
## 7 coopMulti    TRUE        FALSE    20
## 8 coopMulti    TRUE        TRUE      5
```

Demographics

```
d.demographics
```

```
## # A tibble: 268 x 12
##    gameId       createdAt             age gender language    raceWhite raceBlack
##    <chr>        <dttm>              <dbl> <chr>  <chr>       <lgl>     <lgl>
##  1 R6fYnby2Hvs~ 2021-07-20 17:17:52    21 Female English     TRUE      NA
##  2 R6fYnby2Hvs~ 2021-07-20 17:18:14    23 Female English     TRUE      NA
##  3 R6fYnby2Hvs~ 2021-07-20 17:18:27    43 Male   English     TRUE      NA
##  4 KiBJ5D2Gb56~ 2021-07-20 17:26:37    24 Female British En~ TRUE      NA
##  5 KiBJ5D2Gb56~ 2021-07-20 17:26:45    36 Female English     TRUE      NA
##  6 KiBJ5D2Gb56~ 2021-07-20 17:26:54    33 male   English     TRUE      NA
##  7 JvZjCtThY3i~ 2021-07-20 17:40:04    21 Female English     TRUE      NA
##  8 JvZjCtThY3i~ 2021-07-20 17:40:18    47 Female English     TRUE      NA
##  9 JvZjCtThY3i~ 2021-07-20 17:40:47    28 male   English     TRUE      NA
## 10 ier2nCDT3gE~ 2021-07-20 18:22:46    55 female english     TRUE      NA
## # ... with 258 more rows, and 5 more variables: raceAsian <lgl>,
## #   raceNative <lgl>, raceIslander <lgl>, raceHispanic <lgl>, education <chr>
```

```
d.feedback
```

```
## # A tibble: 268 x 8
##    playerId  correctness human workedWell  fair     chatUseful feedback  time
##    <chr>     <chr>       <chr> <chr>       <chr>    <chr>      <chr>     <chr>
##  1 Xzwn3GeD~ yes         yes   disagree    Yes      Yes        My playe~ Yes
##  2 Z4uGdog3~ yes         yes   disagree    The pay~ The chat ~ <NA>      Enough~
##  3 HattoicC~ yes         no    agree       Yes      Yes        No        Yes
##  4 pgdSEje8~ yes         yes   neutral     yes      yes        no        yes
##  5 mQ4j9jv3~ yes         yes   agree       Yes      Very easy  None      Yes
##  6 kq9vcpqm~ yes         yes   stronglyAg~ i think~ yes it wa~ no it wa~ more t~
##  7 nzCnCnx4~ yes         yes   stronglyAg~ Definit~ Yep        <NA>      Perfec~
##  8 WpuqGRTs~ yes         yes   stronglyAg~ The pay~ Very easy. It was a~ yes
##  9 XR254nXR~ yes         yes   stronglyAg~ Yes I do it was ve~ I enjoye~ more t~
## 10 HatzPcAx~ yes         yes   agree       yes      yes        no        yes
## # ... with 258 more rows
```

Basic Analyses

Simple Distributions

```
#get the # of raw points, max adjusted points, game legnth, # of words exchanged
d.round_word_counts <- d.raw_chat.final %>%
  filter(type == "message") %>%
  full_join(d.rounds.final, c("gameId", "trialNum", "repNum",
                              "playerId", "numPlayers", "blockNum")) %>%
 # filter(!is.chitchat) %>%
  mutate(text = gsub("\\n", '', fixed = T, text),
         text = gsub("[/?/.]", ' ', text),
         text = str_squish(text),
         utt_length_chars = str_length(text),
         utt_length_words = str_count(text, "\\W+") + 1) %>%
  group_by(gameId, blockNum, trialNum, repNum, playerId, numPlayers) %>%
  summarize(text = paste0(text, collapse = ', '),
            total_num_words = sum(utt_length_words),
            total_num_chars = sum(utt_length_chars))
```

```
## ‘summarise()‘ has grouped output by ’gameId’, ’blockNum’, ’trialNum’, ’repNum’, ’playerId’. You can
```

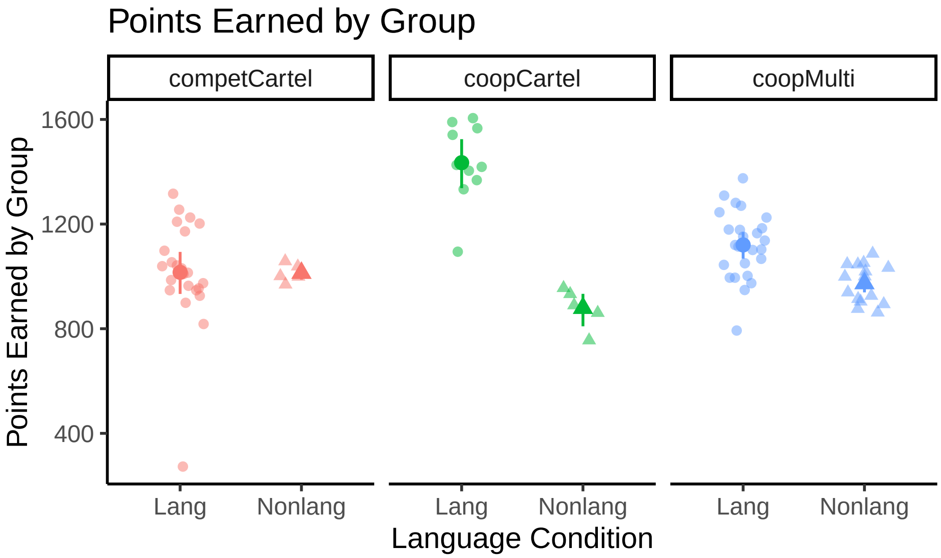
```
d.by_game_metrics <- d.rounds.final%>%
  left_join(d.games %>% select(-createdAt)) %>%
  left_join(d.round_word_counts) %>%
  left_join(game_completion %>% select(gameId, num_rounds)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-")) %>%
  #group by game
  group_by(gameId, name,condition, languageCondition, fullCondition, gameLength,num_rounds) %>%
  summarize(groupPoints = sum(playerUtility, na.rm = T),
            groupNumWords = sum(total_num_words, na.rm = T),
            groupNumChars = sum(total_num_chars, na.rm = T))%>%
  mutate(groupPoints = ifelse(condition=="coopCartel",
                              groupPoints/3, groupPoints)) %>%
  #adjust for max
  mutate(adjustedPoints = case_when(condition == "coopCartel" ~ groupPoints/ ((12*3)*48),
                                    condition == "coopMulti" ~ groupPoints/ ((12+11+10)*48),
                                    condition == "competCartel" ~ groupPoints/ ((12+11+10)*48)))
```

```
## Joining, by = "gameId"

## Joining, by = c("gameId", "repNum", "blockNum", "trialNum", "numPlayers", "playerId")

## Joining, by = "gameId"

## ‘summarise()‘ has grouped output by ’gameId’, ’name’, ’condition’, ’languageCondition’, ’fullConditi
```

Points

Unadjusted

```
d.by_game_metrics%>%
  ggplot(aes(x = languageCondition, y = groupPoints, color = condition, shape = languageCondition)) +
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition)) + labs(x = "Language Condition", y = "Points Earned by Group", titl
```

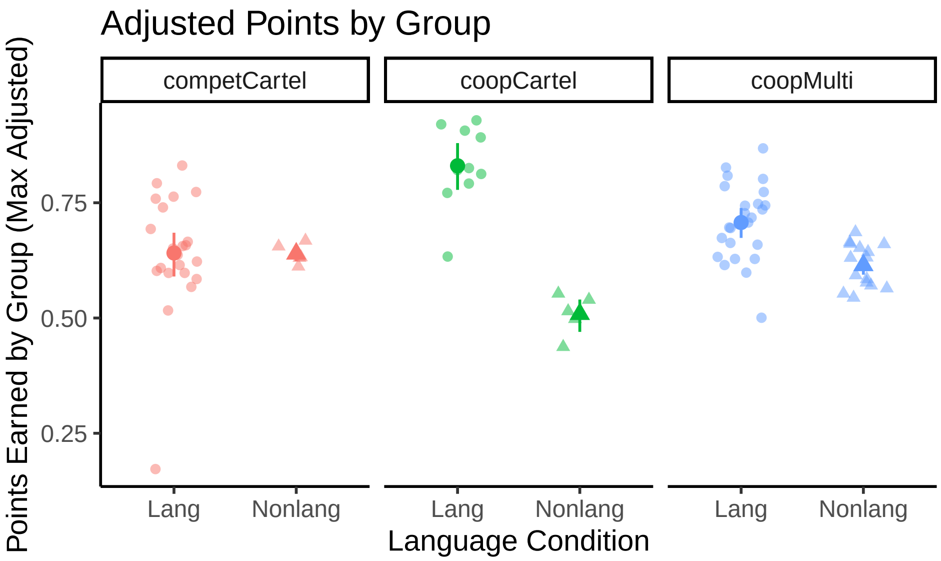


```
ggsave(here(fig_path, "points_distirbution_raw.png"))
```

```
## Saving 5 x 3 in image
```

Max Adjusted

```
d.by_game_metrics %>%
  ggplot(aes(x = languageCondition, y = adjustedPoints, color = condition, shape = languageCondition)) +
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition)) + labs(x = "Language Condition", y = "Points Earned by Group (Max A
```

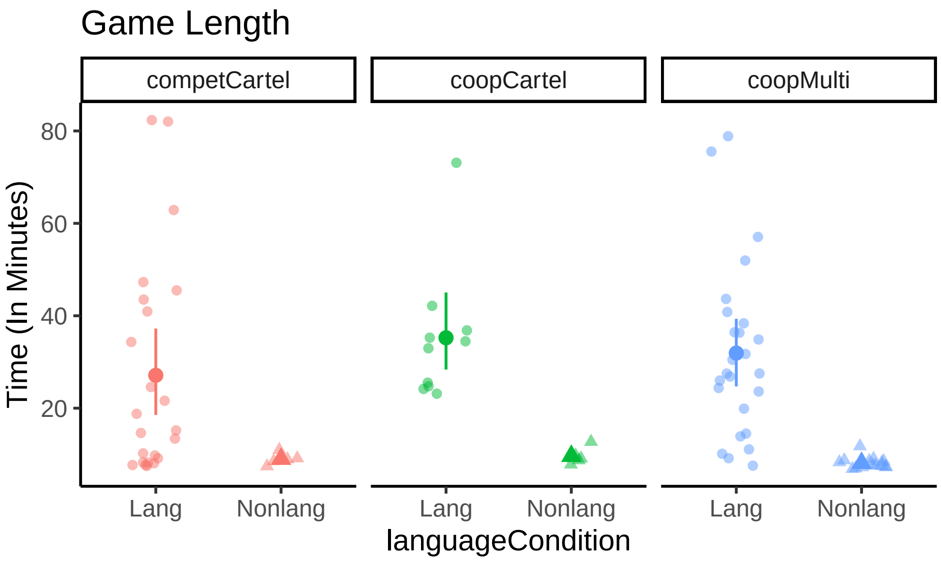


```
ggsave(here(fig_path, "points_distirbution_adjusted.png"))
```

```
## Saving 5 x 3 in image
```

Time

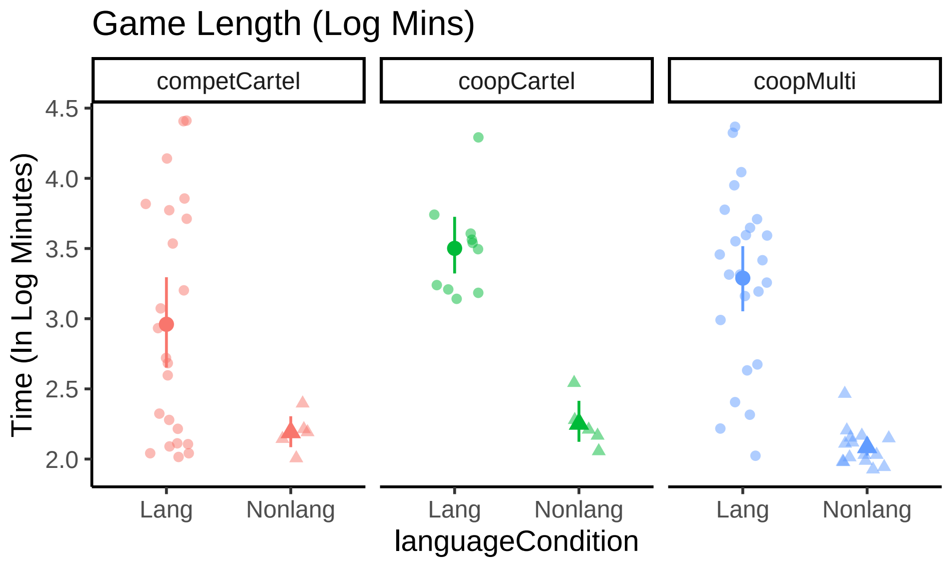
```
d.by_game_metrics %>%
  ggplot(aes(x = languageCondition, y = gameLength/60, color = condition, shape = languageCondition)) +
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition))+ labs(y = "Time (In Minutes)", title = "Game Length")+theme_classi
```



```
ggsave(here(fig_path, "time_distirbution.png"))
```

```
## Saving 5 x 3 in image
```

```
d.by_game_metrics %>%
  ggplot(aes(x = languageCondition, y = log(gameLength/60), color = condition, shape = languageConditio
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition))+ labs(y = "Time (In Log Minutes)", title = "Game Length (Log Mins)
```

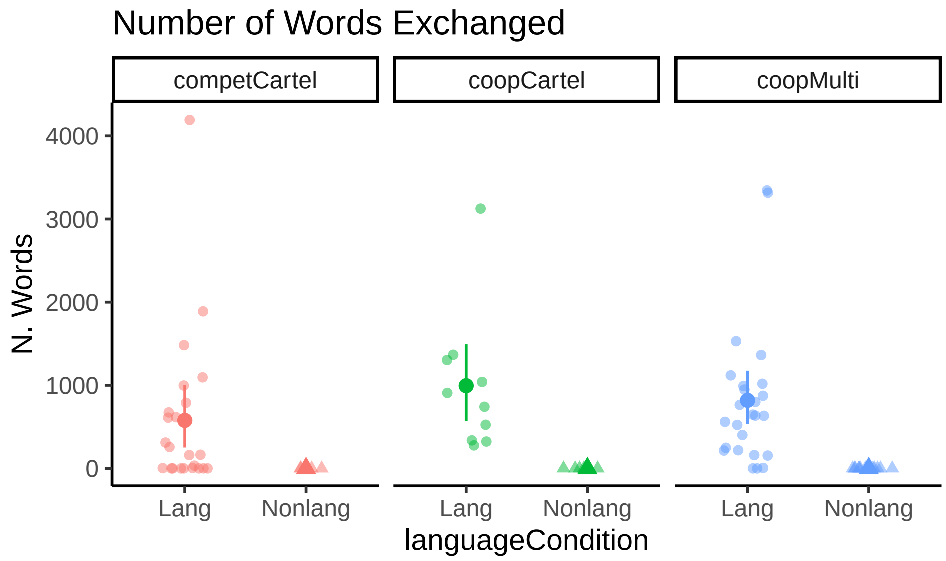


```
ggsave(here(fig_path, "time_log_distirbution.png"))
```

```
## Saving 5 x 3 in image
```

Number of Words

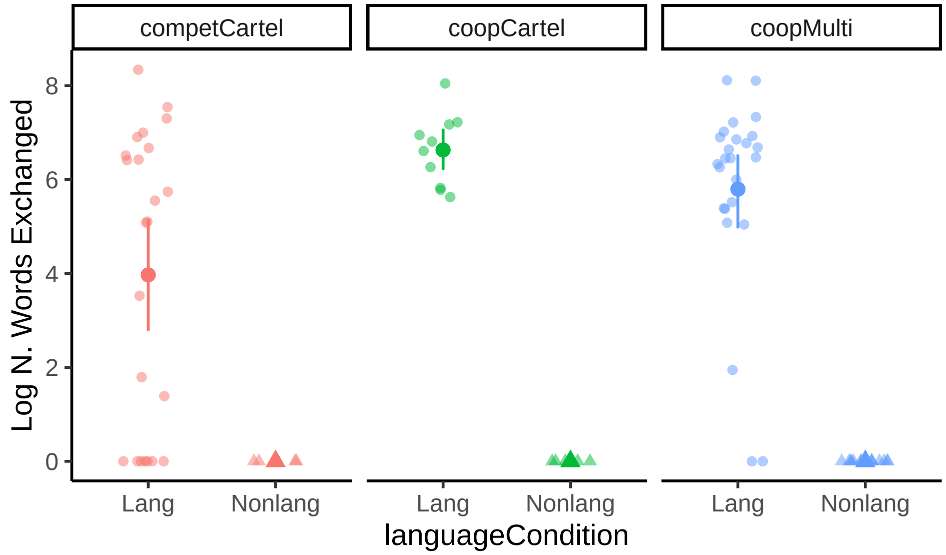
```
d.by_game_metrics %>%
  ggplot(aes(x = languageCondition, y = groupNumWords, color = condition, shape = languageCondition)) +
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition))+ labs(y = "N. Words", title = "Number of Words Exchanged")+theme_c
```



```
ggsave(here(fig_path, "numwords_distirbution.png"))
```

```
## Saving 5 x 3 in image
```

```
d.by_game_metrics %>%
  ggplot(aes(x = languageCondition, y = log(groupNumWords+1), color = condition, shape = languageCondit
  geom_jitter(width = .2, alpha = .5) +
  stat_summary(fun.data = "mean_cl_boot")+
  facet_grid(cols = vars(condition))+ labs(y = "Log N. Words Exchanged")+theme_classic() + theme(legend
```



```
ggsave(here(fig_path, "numwords_log_distirbution.png"))
```

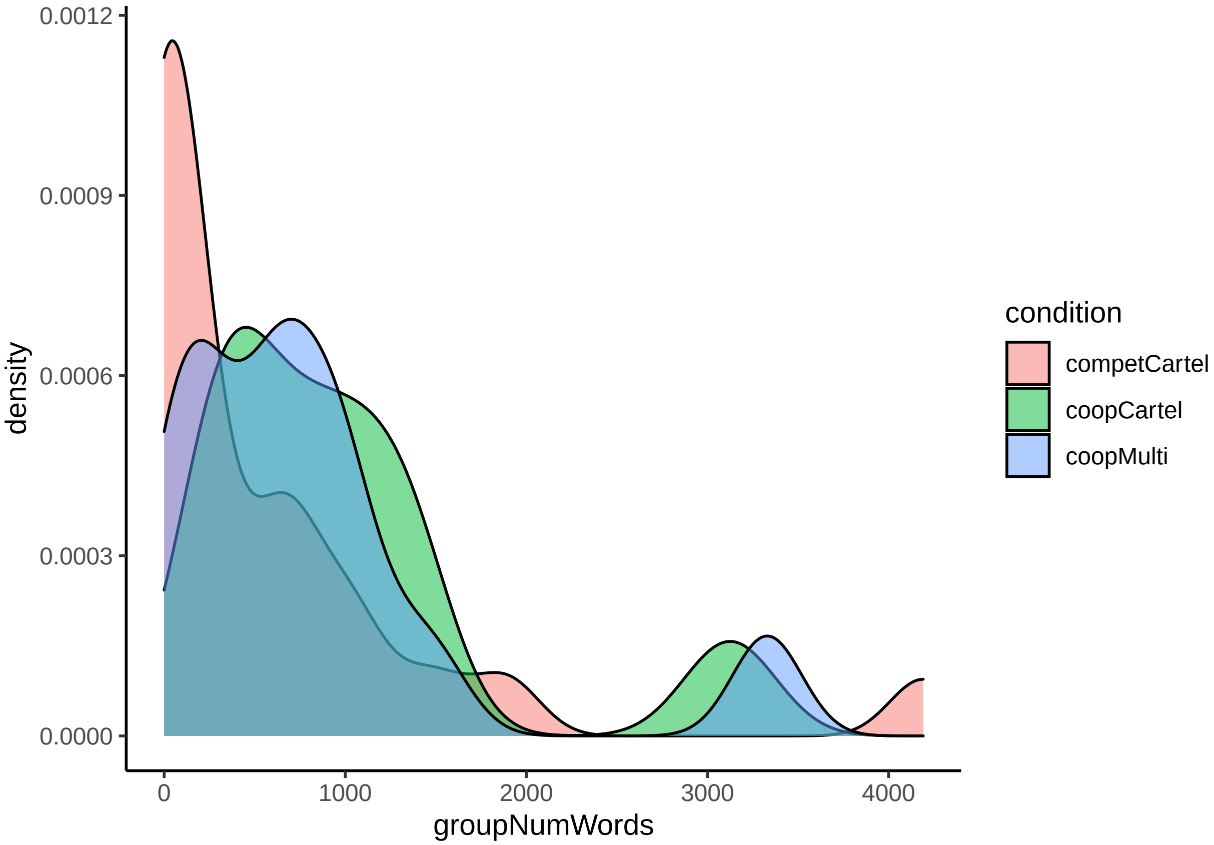
```
## Saving 5 x 3 in image
```

```
d.round_word_counts %>%left_join(d.games %>%
                                   select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(languageCondition == "Lang") %>%
  group_by(gameId, fullCondition, languageCondition, condition) %>%
  summarize(groupNumWords = sum(total_num_words, na.rm = T)) %>%
  ggplot() + geom_density(aes(x = groupNumWords, fill = condition), alpha = .5, adjust = .7)
```

Condition Effects on Language Use

```
## Joining, by = "gameId"
```

```
## ‘summarise()‘ has grouped output by ’gameId’, ’fullCondition’, ’languageCondition’. You can override
```

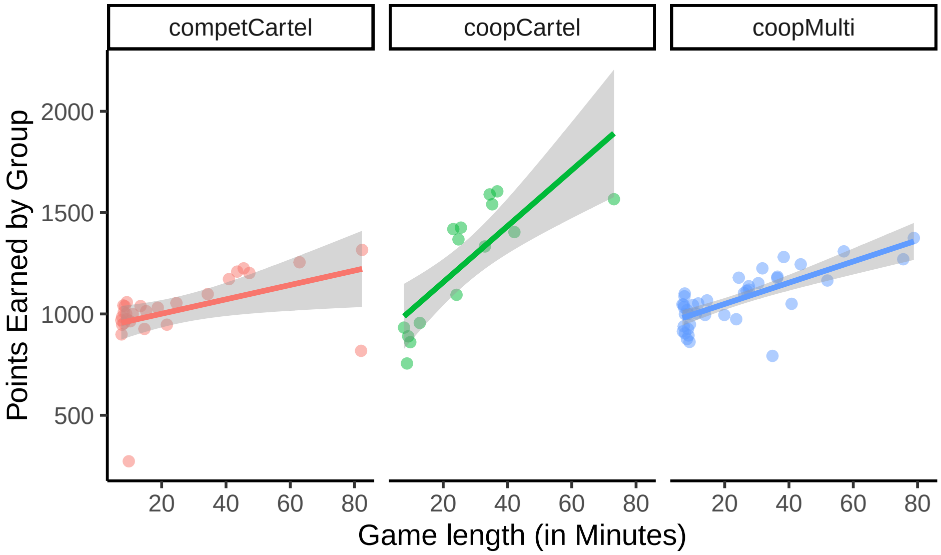


Scatters

Time x Points

```
d.by_game_metrics %>%
  ggplot(aes(x = gameLength/60, y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(condition))+ labs(x = "Game length (in Minutes)", y = "Points Earned by Group"
```

```
## `geom_smooth()` using formula 'y ~ x'
```

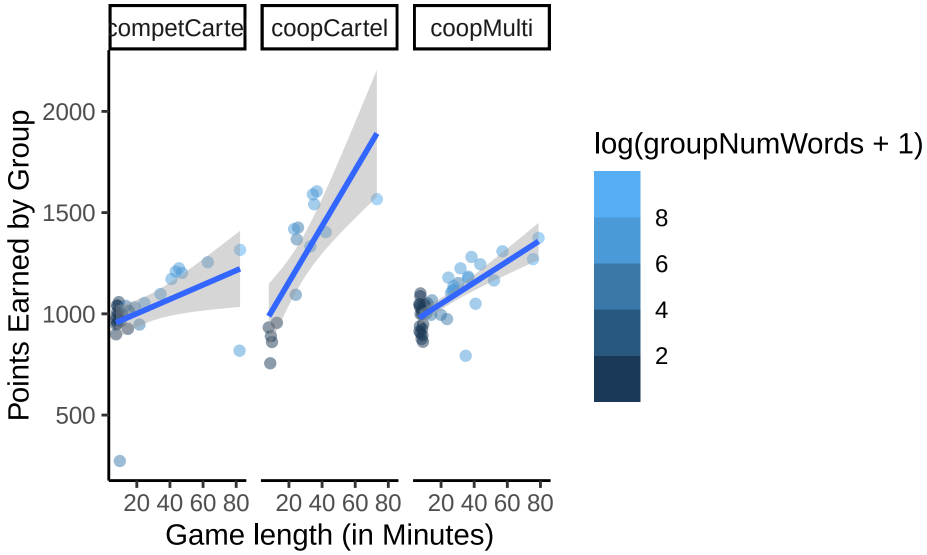


```
ggsave(here(fig_path, "time_points_scatter.png"))
```

```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

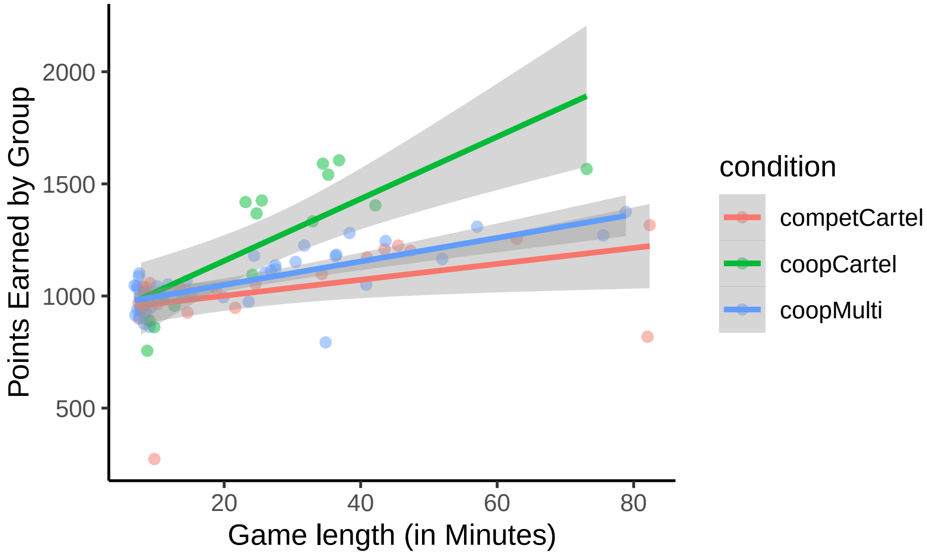
```
d.by_game_metrics %>%
  ggplot(aes(x = gameLength/60, y = groupPoints, color = log(groupNumWords+1))) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(condition))+ labs(x = "Game length (in Minutes)", y = "Points Earned by Group"
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
d.by_game_metrics %>%
  ggplot(aes(x = gameLength/60, y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  labs(x = "Game length (in Minutes)", y = "Points Earned by Group")+theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



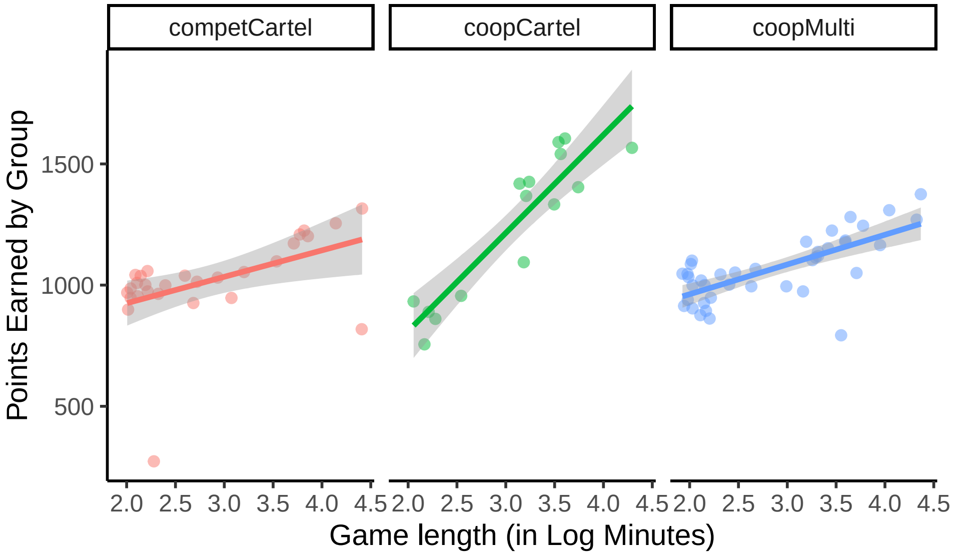
```
ggsave(here(fig_path, "time_points_scatter_nofacet.png"))
```

```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

Log

```
d.by_game_metrics %>%
  ggplot(aes(x = log(gameLength/60), y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(condition))+ labs(x = "Game length (in Log Minutes)",y = "Points Earned by Grou
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

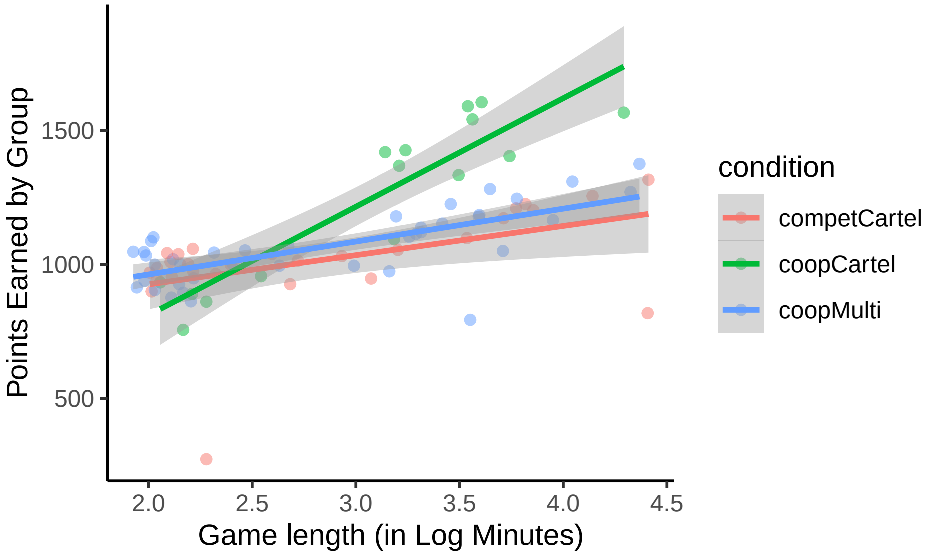


```
ggsave(here(fig_path, "time_points_scatter_facet.png"))
```

```
## Saving 5 x 3 in image
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
d.by_game_metrics %>%
  ggplot(aes(x = log(gameLength/60), y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  labs(x = "Game length (in Log Minutes)",y = "Points Earned by Group")+theme_classic()
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```r
ggsave(here(fig_path, "log_time_points_scatter_nofacet.png"))
```

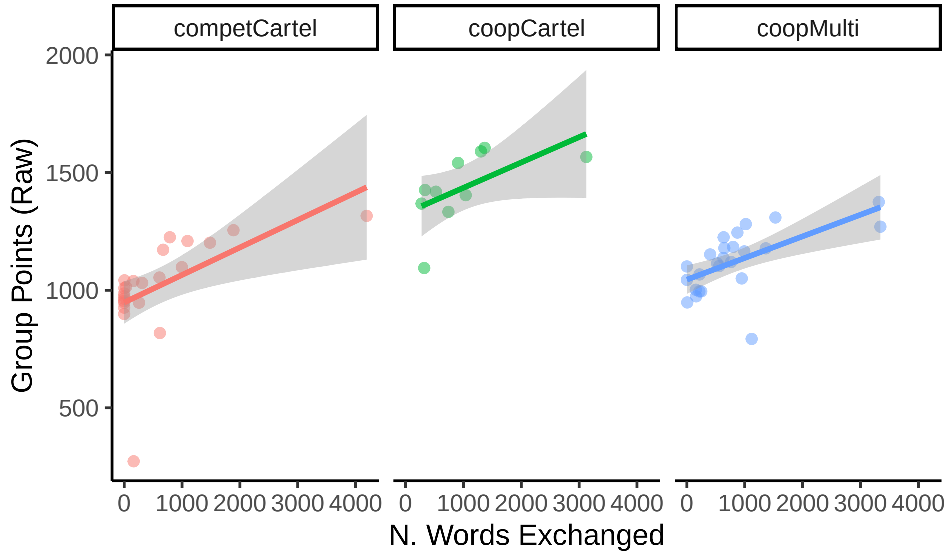
```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

Language x Points

```r
d.by_game_metrics %>% filter(languageCondition == "Lang") %>%
  ggplot(aes(x = groupNumWords, y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(condition))+ labs(x = "N. Words Exchanged", y = "Group Points (Raw)")+theme_cla
```

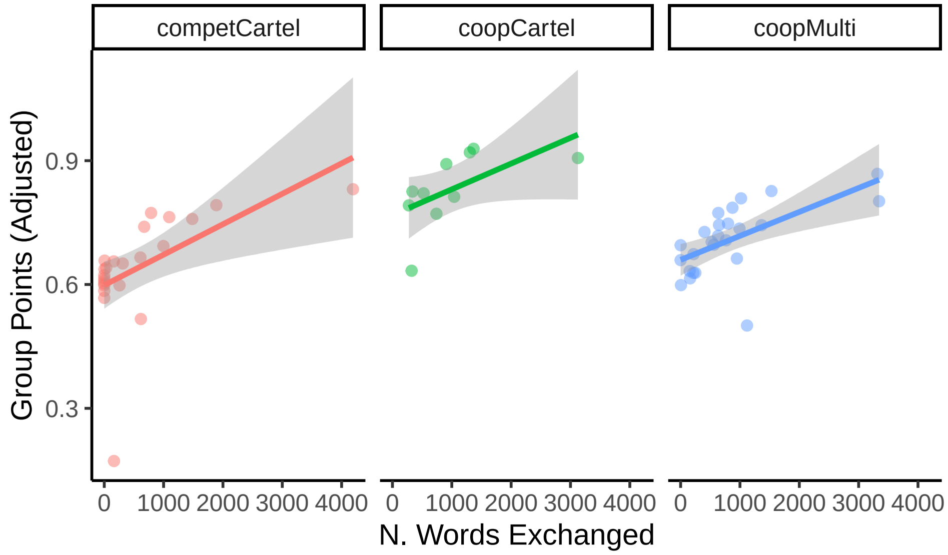
Faceted

```
## `geom_smooth()` using formula 'y ~ x'
```



```
d.by_game_metrics %>% filter(languageCondition == "Lang") %>%
  ggplot(aes(x = groupNumWords, y = adjustedPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(condition))+ labs(x = "N. Words Exchanged", y = "Group Points (Adjusted)")+ther
```

```
## `geom_smooth()` using formula 'y ~ x'
```



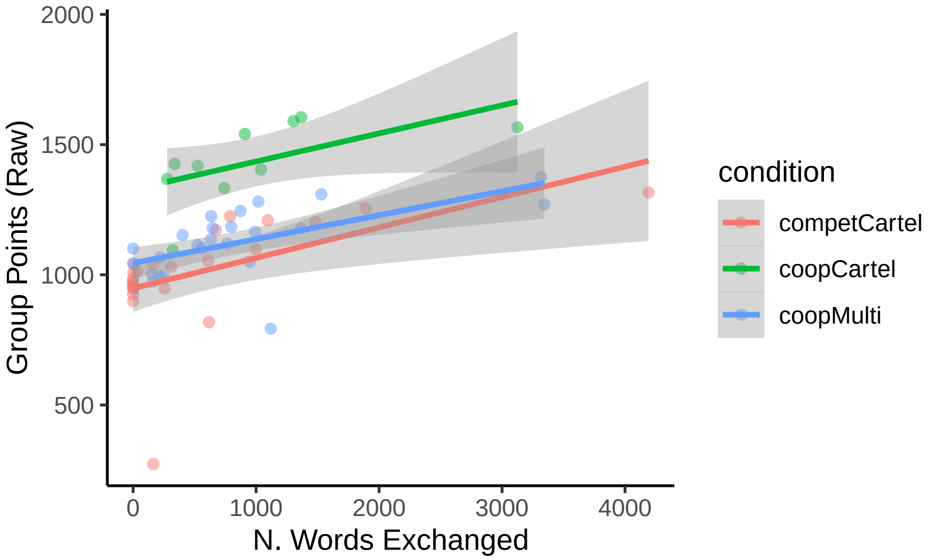
```
ggsave(here(fig_path, "nwords_points_scatter.png"))
```

```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

```
d.by_game_metrics %>% filter(languageCondition == "Lang") %>%
  ggplot(aes(x = groupNumWords, y = groupPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  labs(x = "N. Words Exchanged", y = "Group Points (Raw)")+theme_classic()
```

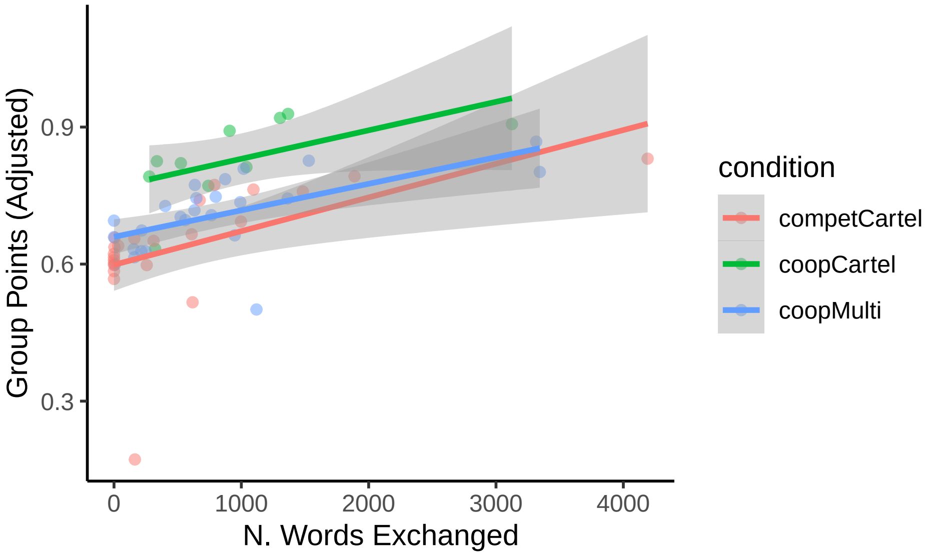
Non-Faceted

```
## `geom_smooth()` using formula 'y ~ x'
```



```
d.by_game_metrics %>% filter(languageCondition == "Lang") %>%
  ggplot(aes(x = groupNumWords, y = adjustedPoints, color = condition)) +
  geom_point(alpha = .5) +
  geom_smooth(method=lm)+
  labs(x = "N. Words Exchanged", y = "Group Points (Adjusted)")+theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave(here(fig_path, "nwords_points_scatter_nofacet.png"))
```

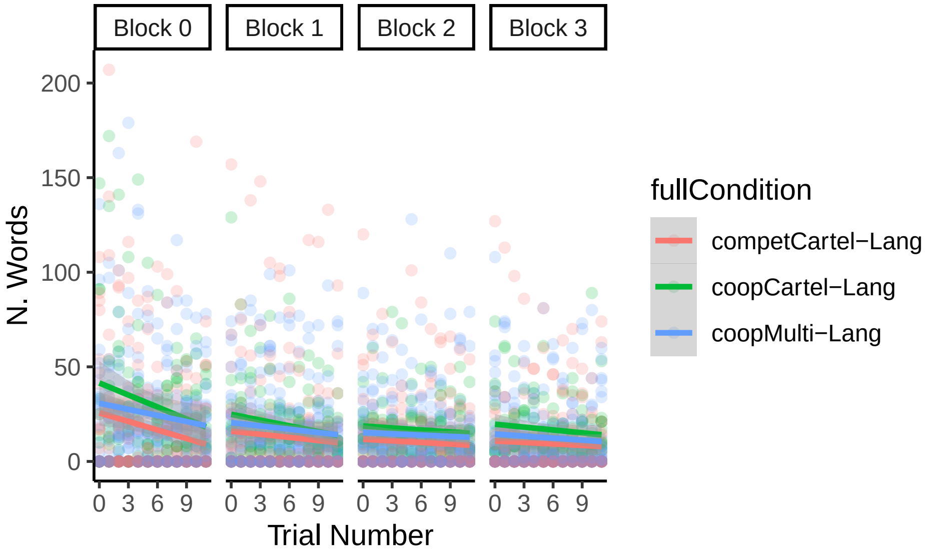
```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

Progress Over Time

Including language optional games where they decided not to speak

```
d.round_word_counts %>%left_join(d.games %>%
                                   select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(languageCondition == "Lang") %>%
  group_by(gameId, fullCondition,block_name, trialNum, repNum) %>%
  summarize(groupNumWords = sum(total_num_words, na.rm = T)) %>%
  ggplot(aes(x = repNum, y = groupNumWords, color = fullCondition)) +
  geom_point(alpha = .2) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(block_name)) + labs(x = "Trial Number", y = "N. Words")+ theme_classic()
```

```
## Joining, by = "gameId"

## `summarise()` has grouped output by 'gameId', 'fullCondition', 'block_name', 'trialNum'. You can ove

## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave(here(fig_path, "nwords_over_time.png"))
```

```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

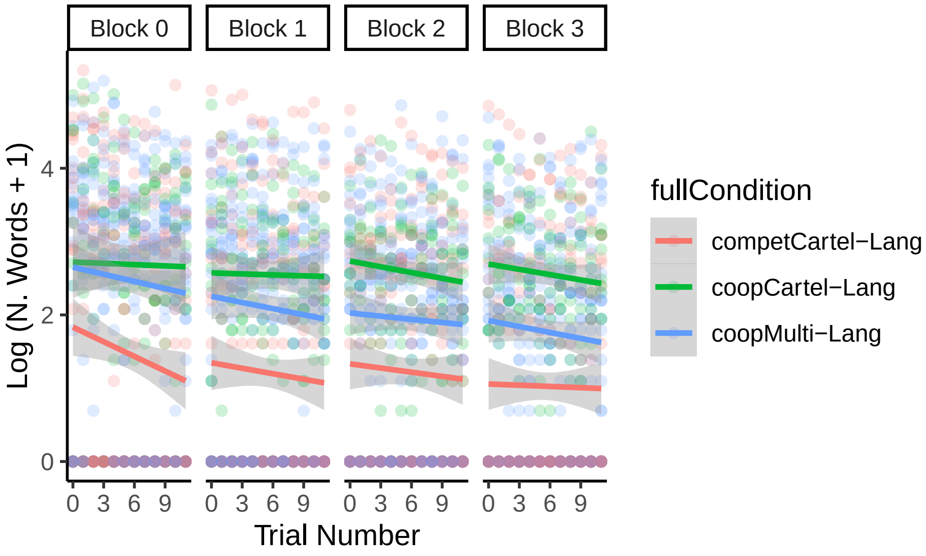
log (n words+1)

```
d.round_word_counts %>%left_join(d.games %>%
                                   select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(languageCondition == "Lang") %>%
  group_by(gameId, fullCondition,block_name, trialNum, repNum) %>%
  summarize(groupNumWords = sum(total_num_words, na.rm = T)+1) %>%
  ggplot(aes(x = repNum, y = log(groupNumWords), color = fullCondition)) +
  geom_point(alpha = .2) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(block_name)) + labs(x = "Trial Number", y = "Log (N. Words + 1)")+ theme_class
```

```
## Joining, by = "gameId"
```

```
## `summarise()` has grouped output by 'gameId', 'fullCondition', 'block_name', 'trialNum'. You can ove
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave(here(fig_path, "nwords_log_over_time.png"))
```

```
## Saving 5 x 3 in image
## `geom_smooth()` using formula 'y ~ x'
```

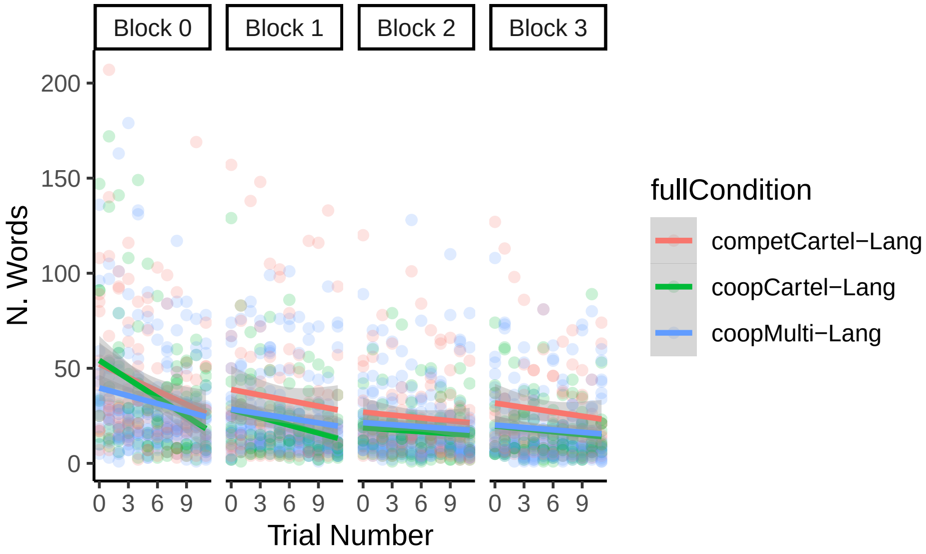
Only games where language was used

```
d.round_word_counts %>%left_join(d.games %>%
                                   select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(languageCondition == "Lang") %>%
  group_by(gameId, fullCondition,block_name, trialNum, repNum) %>%
  summarize(groupNumWords = sum(total_num_words, na.rm = T)) %>%
  filter(groupNumWords > 0) %>%
  ggplot(aes(x = repNum, y = groupNumWords, color = fullCondition)) +
  geom_point(alpha = .2) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(block_name)) + labs(x = "Trial Number", y = "N. Words")+ theme_classic()
```

```
## Joining, by = "gameId"
```

```
## `summarise()` has grouped output by 'gameId', 'fullCondition', 'block_name', 'trialNum'. You can ove
```

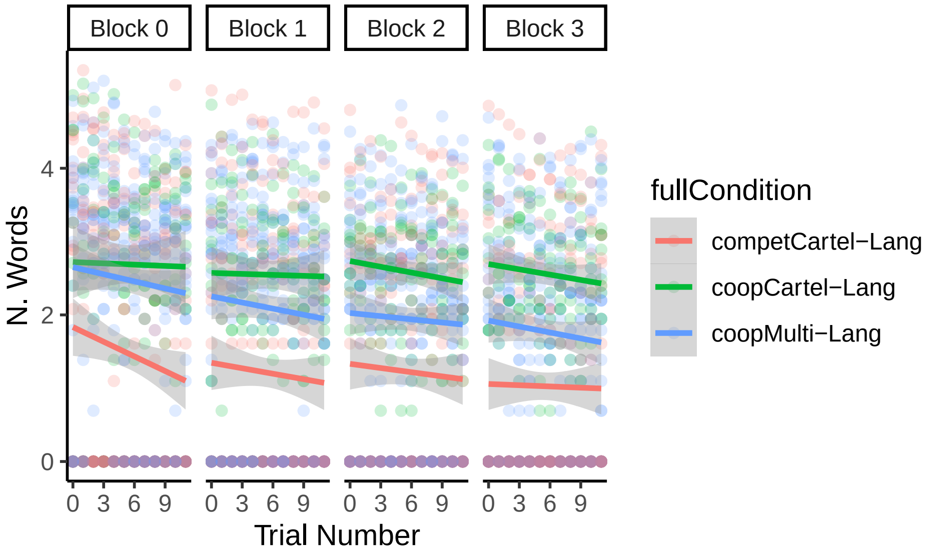
```
## `geom_smooth()` using formula 'y ~ x'
```



log num words

```
d.round_word_counts %>%left_join(d.games %>%
                                   select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(languageCondition == "Lang") %>%
  group_by(gameId, fullCondition,block_name, trialNum, repNum) %>%
  summarize(groupNumWords = sum(total_num_words, na.rm = T)) %>%
  ggplot(aes(x = repNum, y = log(groupNumWords+1), color = fullCondition)) +
  geom_point(alpha = .2) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(block_name)) + labs(x = "Trial Number", y = "N. Words")+ theme_classic()
```

```
## Joining, by = "gameId"

## `summarise()` has grouped output by 'gameId', 'fullCondition', 'block_name', 'trialNum'. You can ove

## `geom_smooth()` using formula 'y ~ x'
```



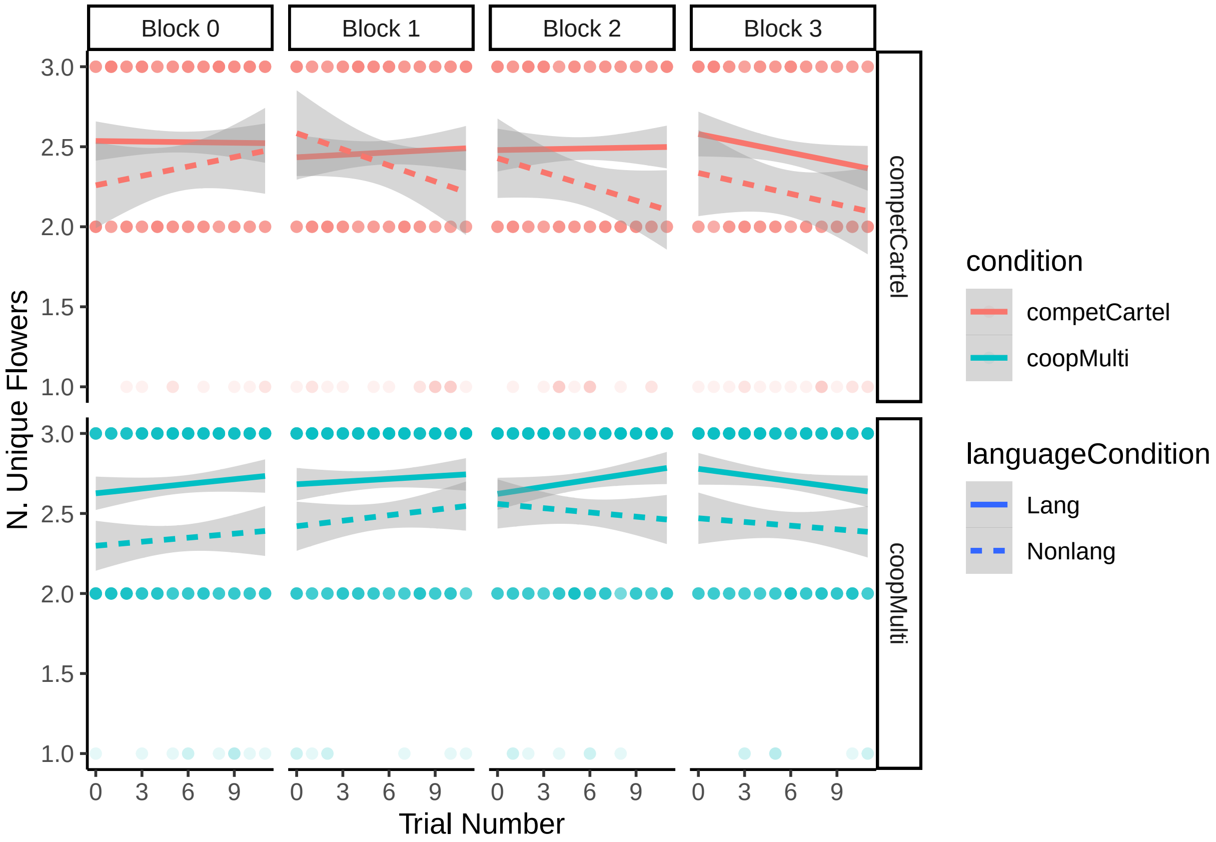
### Accuracy #### N. Flowers Selected each round

```
d.rounds.final %>% group_by(gameId, repNum, blockNum, trialNum) %>%
  summarize(num_unique_flowers = n_distinct(playerResponse),
            mean_response_time = mean(time_sec, na.rm = T))  %>%
  left_join(d.games %>% select(gameId, condition,chatEnabled)) %>%
  mutate(languageCondition = ifelse(chatEnabled, "Lang", "Nonlang"),
         fullCondition = paste(condition, languageCondition, sep = "-"),
         block_name = paste0("Block ", blockNum)) %>%
    filter(condition != "coopCartel") %>%
  ggplot(aes(x = repNum, y = num_unique_flowers, color = condition, linetype = languageCondition)) +
  geom_point(alpha = .1) +
  geom_smooth(method=lm)+
  facet_grid(cols = vars(block_name), rows = vars(condition)) + labs(x = "Trial Number", y = "N. Unique
```

```
## ‘summarise()‘ has grouped output by ’gameId’, ’repNum’, ’blockNum’. You can override using the ‘.grou
```

```
## Joining, by = "gameId"
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```
# d.round_results.final %>% left_join(d.games %>% select(gameId, condition, chatEnabled, langUse, gameC
#   mutate(chat = ifelse(chatEnabled, "lang", "nonlang"),
#          full_condition = paste0(condition, "-", chat)) %>%
#   ggplot(aes(x=time_sec, y = as.numeric(playerUtility), color = full_condition)) +
#   facet_grid(rows = vars(condition), cols = vars(langUse)) +
#   geom_point(alpha = .25) +
#   labs(x = "log time (in seconds)",
#        y = "individual utility",
#        title = "Response time in sec and utility", subtitle =  "each participant, each round")
#
# d.round_results.final %>% left_join(d.games %>% select(gameId, condition, chatEnabled, langUse)) %>%
#   mutate(chat = ifelse(chatEnabled, "lang", "nonlang"),
#          full_condition = paste0(condition, "-", chat)) %>%
#   ggplot(aes(x=log(time_sec), y = as.numeric(playerUtility), color = full_condition)) +
#   facet_grid(rows = vars(condition), cols = vars(langUse)) +
#   geom_point(alpha = .25) +
#   labs(x = "log time (in seconds)",
#        y = "individual utility",
#        title = "Response time in sec (log) and utility", subtitle =  "each participant, each round")
```

Accuracy sum utility for entire group/max possible utility per group (or, the max flower *3) Each condition is going to have a different maximum utility

coopcartel will have a max utility = max flower * 3 (players) * 3 (incentive)

competcartel will have a max utility = sum(top_3_flowers)

```
# d.max_utility <- d.contexts %>%
#   left_join(d.games %>% select(gameId, condition)) %>%
#   mutate(utility= as.numeric(utility)) %>%
#   group_by(gameId, blockNum, repNum) %>%
#   slice_max(order_by =utility,n = 3)%>%
#   summarize(competCartel = sum(utility),
#             coopMulti= sum(utility),
#             coopCartel = max(utility * 9)) %>% ungroup() %>%
#   pivot_longer(cols = c(competCartel, coopMulti, coopCartel), names_to = "condition", values_to = "ma
#
#
# d.utility <- d.round_results.final %>%
#          left_join(d.games %>% select(gameId, condition, chatEnabled, langUse)) %>%
#   group_by(gameId, blockNum, repNum, trialNum,condition, langUse) %>%
#   summarize(group_utility = sum(as.numeric(playerUtility)))%>% ungroup() %>%
#   left_join(d.max_utility) %>%
#   mutate(prop_utility = group_utility/maxUtility,
#          langUsed = if_else(langUse, "Lang", "Nonlang"))
#
# d.utility %>%
#   ggplot(aes(x = trialNum, y=prop_utility, color = as.factor(gameId))) +
#   geom_point(alpha = .4) +
#   facet_grid(cols = vars(condition), rows = vars(langUsed), scales = "free_y") +
#   #geom_smooth(method=glm, formula=y~poly(x,2), alpha=.3)+
#   theme(legend.position = "none")+
#   labs(title="Accuracy", y="Group utility/max utility", x="Round number", color="gameId")
#
# ggsave(paste0(image_location, "/accuracy.png"))
#
# d.utility %>%
#   ggplot(aes(x = trialNum, y=prop_utility, color = as.factor(gameId))) +
#   geom_point(alpha = .4) +
#   facet_grid(cols = vars(condition), rows = vars(langUsed), scales = "free_y") +
#   #geom_smooth(method=glm, formula=y~poly(x,2), alpha=.3)+
#   theme(legend.position = "none")+
#   labs(title="Accuracy", y="Group utility/max utility", x="Round number", color="gameId")
#
# ggsave(paste0(image_location, "/accuracy.png"))
#
# d.utility %>%
#   filter(langUsed == "Lang") %>%
#   ggplot(aes(x = repNum, y=prop_utility, color = as.factor(gameId))) +
#   geom_point(alpha = .4) +
#   facet_grid(cols = vars(blockNum), rows = vars(gameId)) +
#   #geom_smooth(method=glm, formula=y~poly(x,2), alpha=.3)+
#   theme(legend.position = "none")+
#   labs(title="Accuracy", y="Group utility/max utility", x="Round number", color="gameId") + ylim(0,1)
#
# ggsave(paste0(image_location, "/accuracy.png"))
```

Everything here has bootstrapped 95% CIs.

Should find better curves to fit, but using quadratic to allow for some curvature.

```
# ggplot(d.chat, aes(x=repNum, y=total_num_words, color=role))+
#   facet_wrap(~tangram, nrow=2)+
#   scale_color_brewer(palette="Dark2")+
#      stat_summary(fun.data = "mean_cl_boot")+
#   labs(title="Number of words", y="Number of words", x="Round number")+
#   theme(legend.position="bottom")
# ggplot(d.chat, aes(x=repNum, y=total_num_words, color=as.factor(numPlayers)))+
#   facet_wrap(~role, nrow=1)+
#   scale_color_brewer(palette="Dark2")+
#     geom_jitter(alpha=.05)+
#     geom_smooth(method=glm, formula=y~poly(x,2), alpha=.3)+
#   #geom_smooth(method = "glm", formula = y~x,method.args = list(family = gaussian(link = 'log')))+
#      stat_summary(fun.data = "mean_cl_boot")+
#     scale_y_continuous(limits = c(0,50))+
#   labs(title="Number of words", y="Number of words", x="Round number", color="Player count")+
#   theme(legend.position="bottom")
# ggsave(here(image_location, 'words.pdf'), width=6, height=4)
```

```
# d.chat %>% filter(role=="speaker") %>%
#     mutate(groupxtangram=str_c(gameId,tangram)) %>%
#   group_by(repNum, numPlayers, gameId,tangram, groupxtangram) %>%
#   summarize(words=sum(total_num_words)) %>%
# ggplot(aes(x=repNum, y=words, color=as.factor(numPlayers)))+
#   facet_wrap(~numPlayers, nrow=1)+
#   scale_color_brewer(palette="Dark2")+
#     geom_line(aes(group=groupxtangram), alpha=.1,method=glm, se=F)+
#     geom_smooth(method = "glm", formula = y~x,method.args = list(family = gaussian(link = 'log')))+
#     #geom_smooth(method=glm, formula=y~poly(x,2), alpha=.3)+
#   labs(title="Words from speaker per tangram", y="Number of words", x="Round number", color="Player c
#   theme(legend.position="null")
# ggsave(here(image_location, 'words_lines.pdf'), width=6, height=4)
```

```
# d.chat %>% filter(role=="speaker") %>%
# ggplot(aes(x=repNum, y=total_num_words, color=as.factor(numPlayers)))+
#   facet_wrap(~tangram)+
#   scale_color_brewer(palette="Dark2")+
#     #geom_smooth(method=glm, formula=y~poly(x,2), se=T, alpha=.1)+
#       geom_smooth(method = "glm", formula = y~x,method.args = list(family = gaussian(link = 'log')))+
#        stat_summary(fun.data = "mean_cl_boot", size=.2)+
#   labs(title="Tangram variability", y="Number of words", x="Round number", color="Player count")+
#   theme(legend.position="bottom")
# ggsave(here(image_location, 'words_tangrams.pdf'), width=8, height=6)
```

```
# d.round_results %>% group_by(playerId,repNum, gameId, numPlayers) %>%
#   mutate(correct.num=ifelse(correct,1,0)) %>%
#   ggplot(aes(x=repNum, y=correct.num, color=as.factor(numPlayers)))+
# geom_smooth(method = "glm", method.args = list(family = "binomial")) +
#   stat_summary(fun.data = "mean_cl_boot", position = position_dodge(width=.2))+
#   #geom_point()+
#   scale_color_brewer(palette="Dark2")+
#   #scale_y_continuous(limits = c(0,1))+
#   labs(x="Round Number", y="Fraction correctly selected", title= "Overall accuracy increases over rep
#     theme(legend.position="bottom")
# ggsave(here(image_location, 'accuracy.pdf'), width=6, height=4)
```

```
# d.round_results %>% group_by(playerId, repNum, gameId, numPlayers) %>%
#   filter(correct==T) %>%
#   #summarize(time=mean(time)) %>%
#   ggplot(aes(x=repNum, y=time, color=as.factor(numPlayers)))+
#   geom_jitter(width=.4, height=0, alpha=.03)+
# geom_smooth(method = "glm", formula = y~x,
#                       method.args = list(family = gaussian(link = 'log')))+
#   stat_summary(fun.data = "mean_cl_boot", position = position_dodge(width=.2))+
#   scale_y_continuous(limits = c(0,180))+
#     scale_color_brewer(palette="Dark2")+
#   labs(x="Round Number", y="Time to selection in seconds",
#        title="People choose faster in later rounds", color="Player count")+
#   theme(legend.position = "bottom")
# ggsave(here(image_location, 'time.pdf'), width=6, height=4)
```

Models

```
#summary(model)
```

```
#summary(model_speaker_acc)
```