MP-Game-5: Multiplayer Boolean Concept Learning (Summurization Format)

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
```

```
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(Inf, Inf, Inf, Inf, Inf, :
## invalid factor level, NA generated

## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
```

```
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_character(),
##   role = col_character(),
##   rule_idx = col_character(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 2)
```

```
## Warning: 1 parsing failure.
## row # A tibble: 1 x 5 col     row col   expected   actual    file
```

```
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
```

```
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
```

```
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
## Parsed with column specification:
## cols(
##   iterationName = col_character(),
##   gameid = col_character(),
##   time = col_double(),
##   role = col_character(),
##   rule_idx = col_integer(),
##   rule_type = col_character(),
##   train_data_fn = col_character(),
##   test_data_fn = col_character(),
##   text = col_character(),
##   reactionTime = col_integer(),
##   eventType = col_character()
## )
```

Intro

Our goal is to ollect cheaper data for cultural ratchet experiment. In the spring, we ran a pilot experiment, with a "round-by-round" format, where players only learned a single concept. In fall, we ran a pilot experiment with a "summurization" format, where palyers played multiple concepts. For spring pilot (round-by-round),

we had 5 concepts, each with two different lists of stimuli. For fall pilot (summurization), we had the same 5 concepts and selected one of two lists of stimuli, from spring.

Cost Comparison

Spring Pilot Cost Per Round (Round-By-Round-Format):

$122.40 / 36 = $3.40 / round

Fall Pilot Cost Per Round (Summurization-Format):

($(1.25 * 20 + .50 * 20) + $11.60)) / 40 = $1.165 / round

If we were maximally efficient, the Fall Pilot would be $.892 / round. We had two games crash midway because of players disconnecting; so our cost for the pilot was slightly higher than expected according to the data we collected.

Experiment Details (Critters)

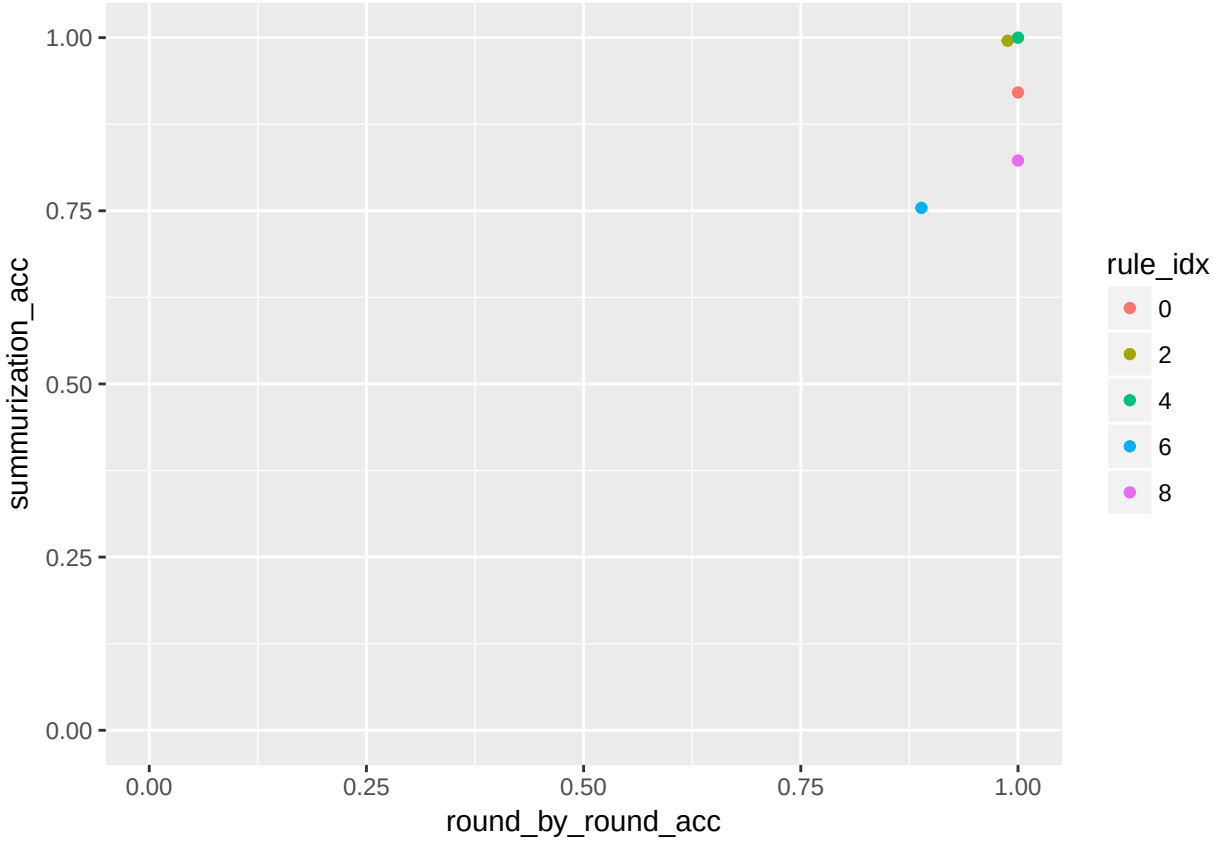
- 81 total possible critters: ~50 training, ~31 test
- 4 axes of variability:
  - Critter Type (Bug, Fish, Bird)
  - Primary Color (Blue, Green, Orange)
  - Secondary Color (Red, Yellow, Purple)
  - Size (Small, Medium, Large)

Experiment Details (Concepts)

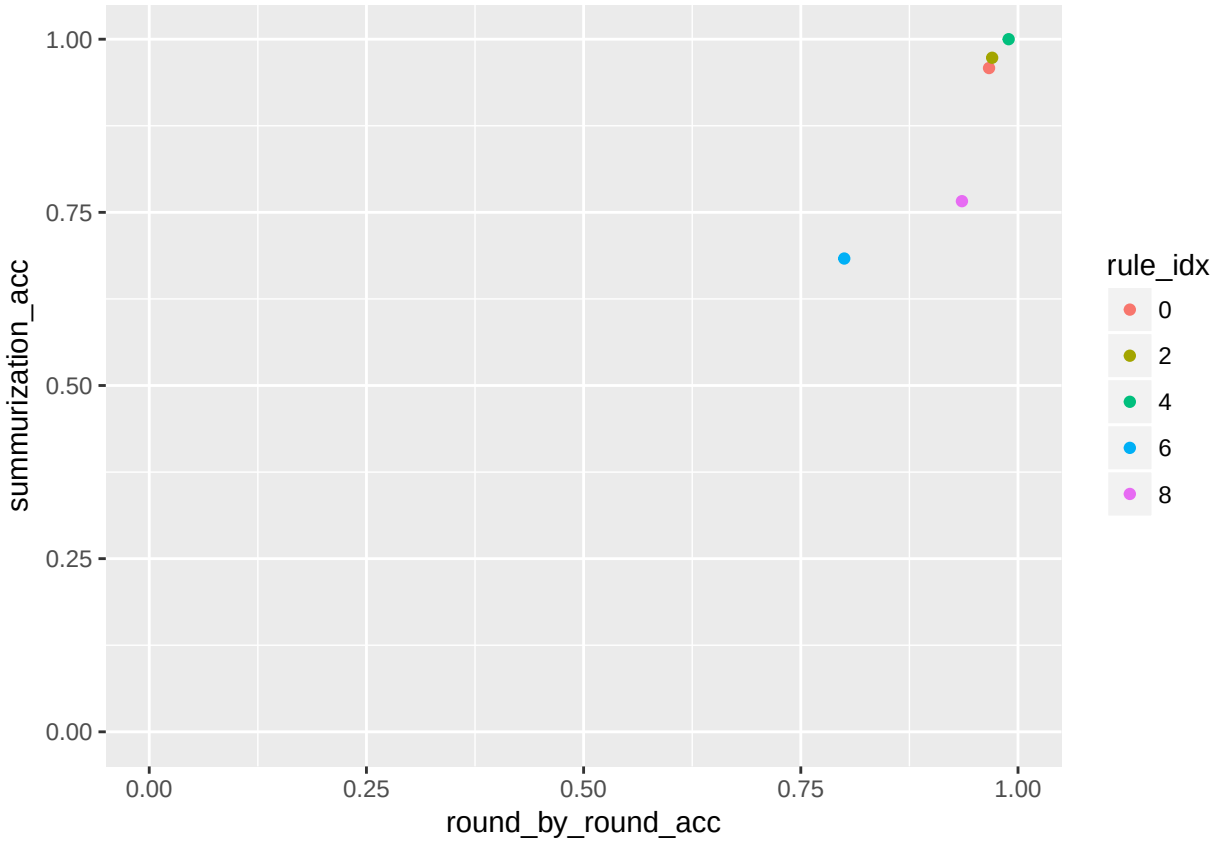
- Rule Idx 0: Primary Color == Orange ("Orange things")
- Rule Idx 2: Critter Type == Fish && Primary Color == Blue ("Blue fish")
- Rule Idx 4: Primary Color == Orange && Secondary Color == Purple ("Purple and orange things")
- Rule Idx 6: Critter Type == Bug || Secondary Color == Yellow ("Bugs, or yellow things")
- Rule Idx 8: Critter Type == Bird || Primary Color == Green ("Birds, or green things")

Round vs. Summurization

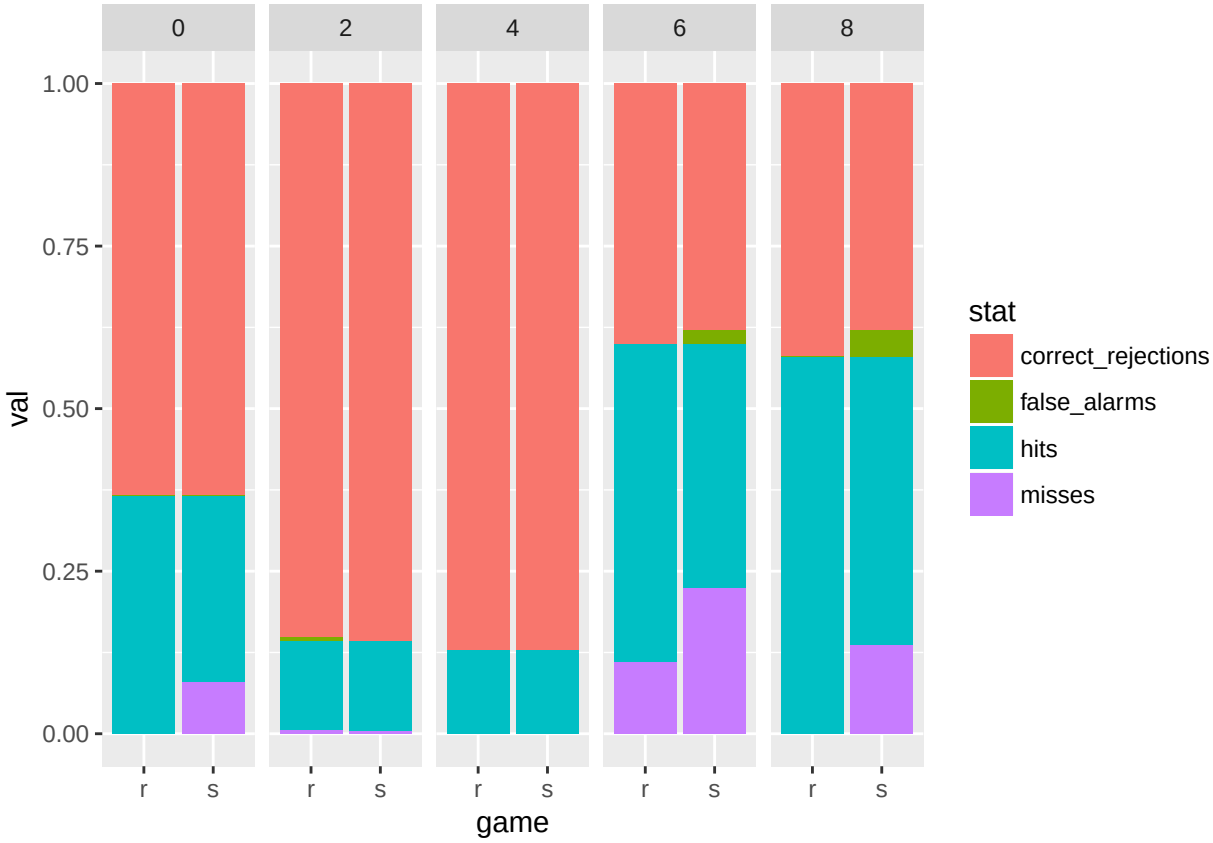
Teacher Performance (Round vs. Summurization Acc.)



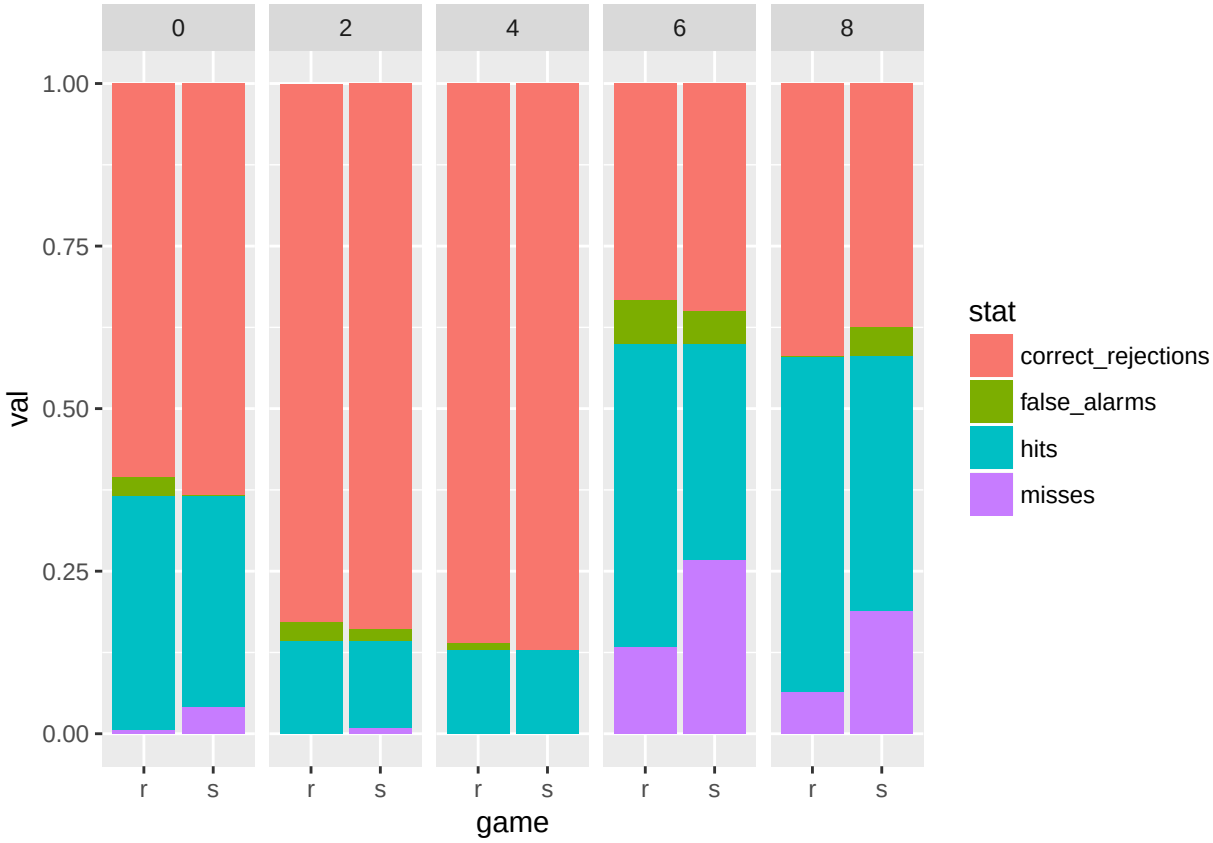
Student Performance (Round vs. Summurization Acc.)



Teacher Performance (Round vs. Summurization)

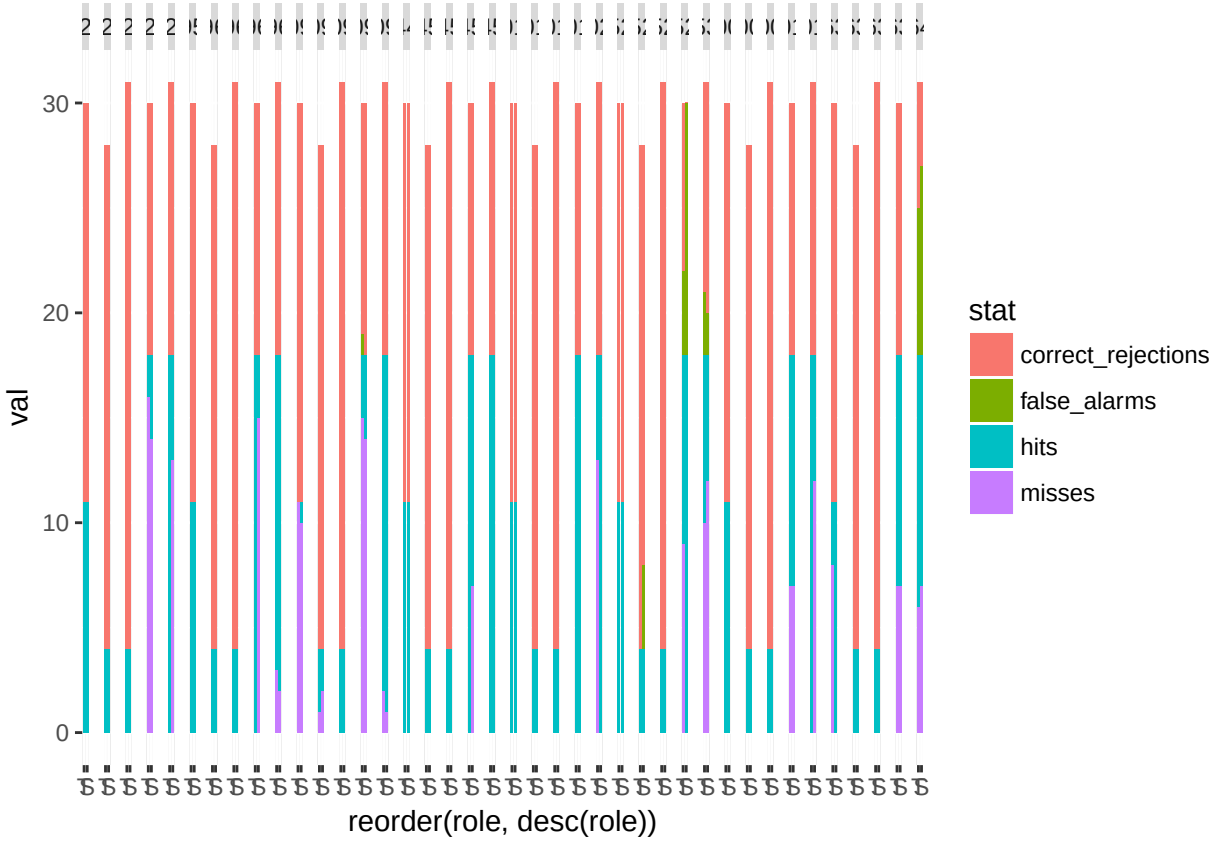


Student Performance (Round vs. Summurization)

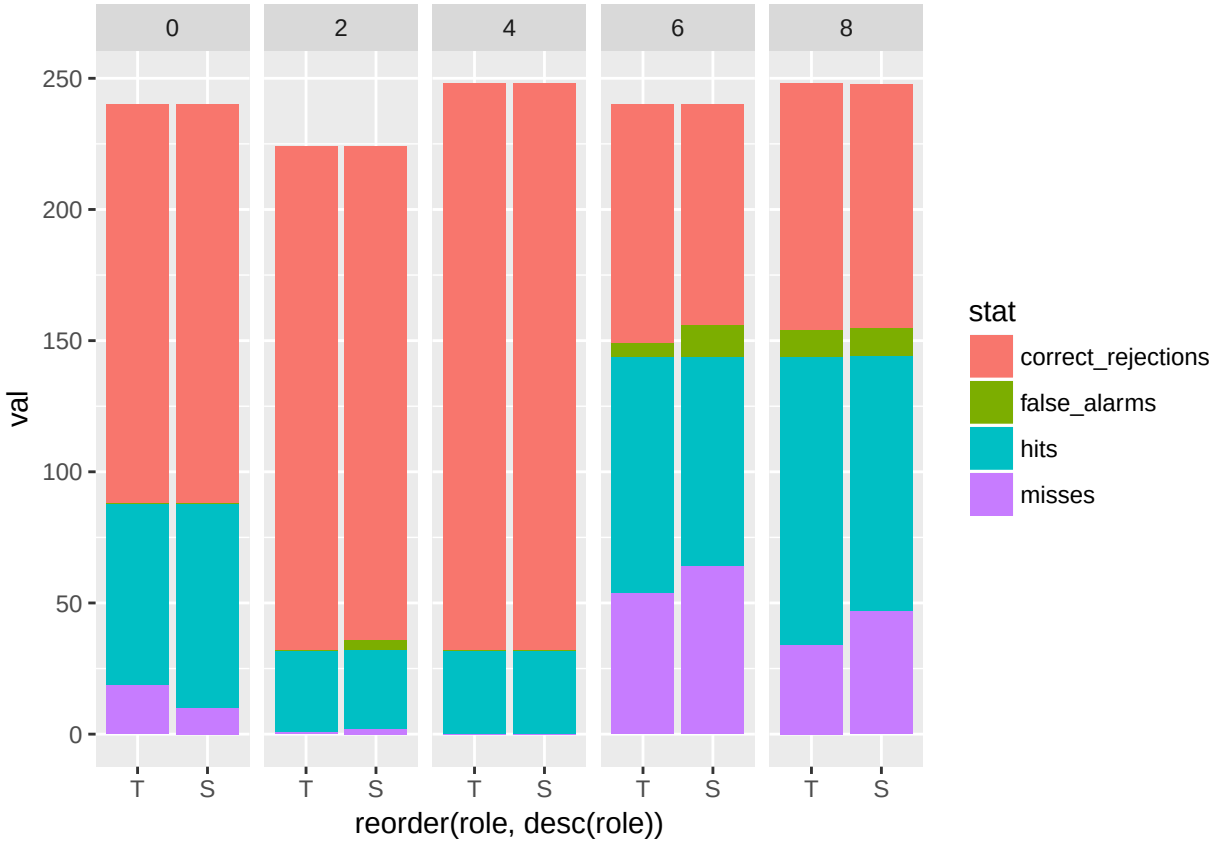


Summurization Game Analysis

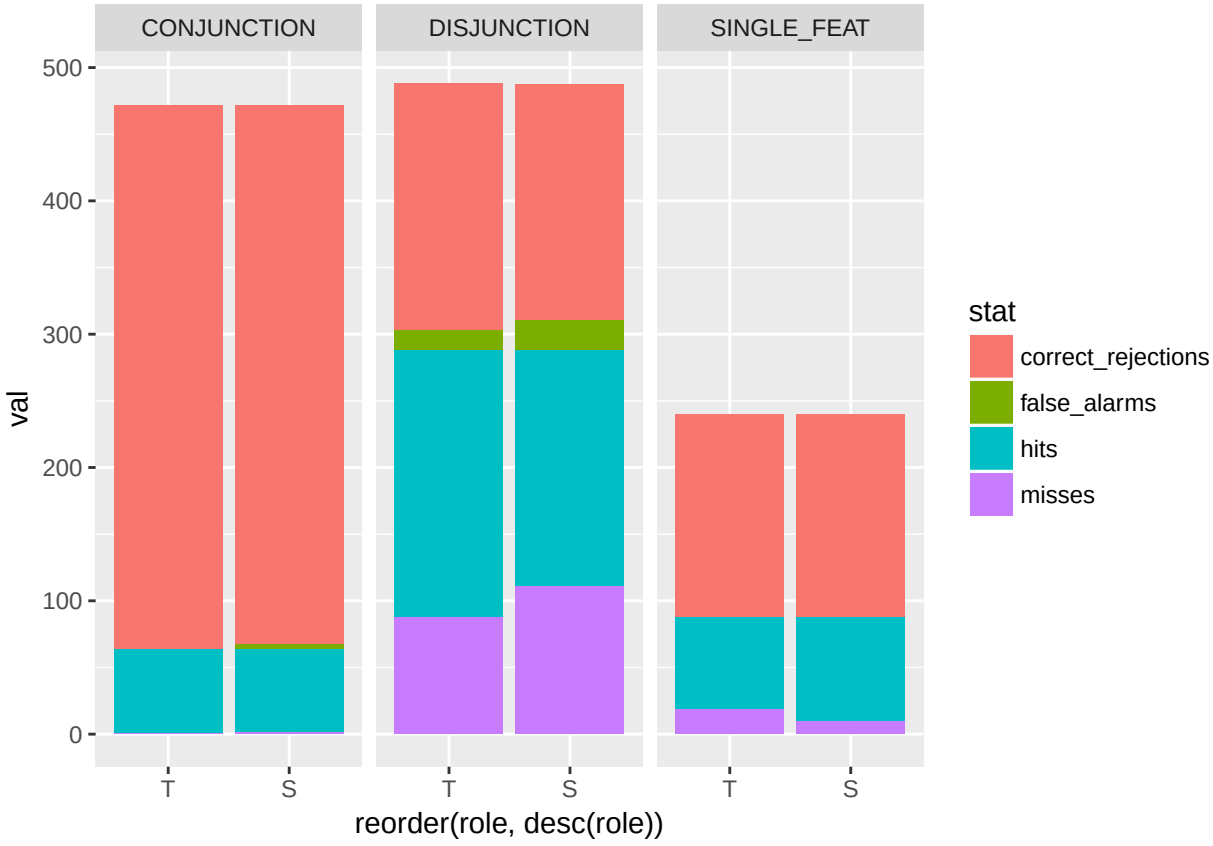
Student versus Teacher Performance – Pairs (Summurization)



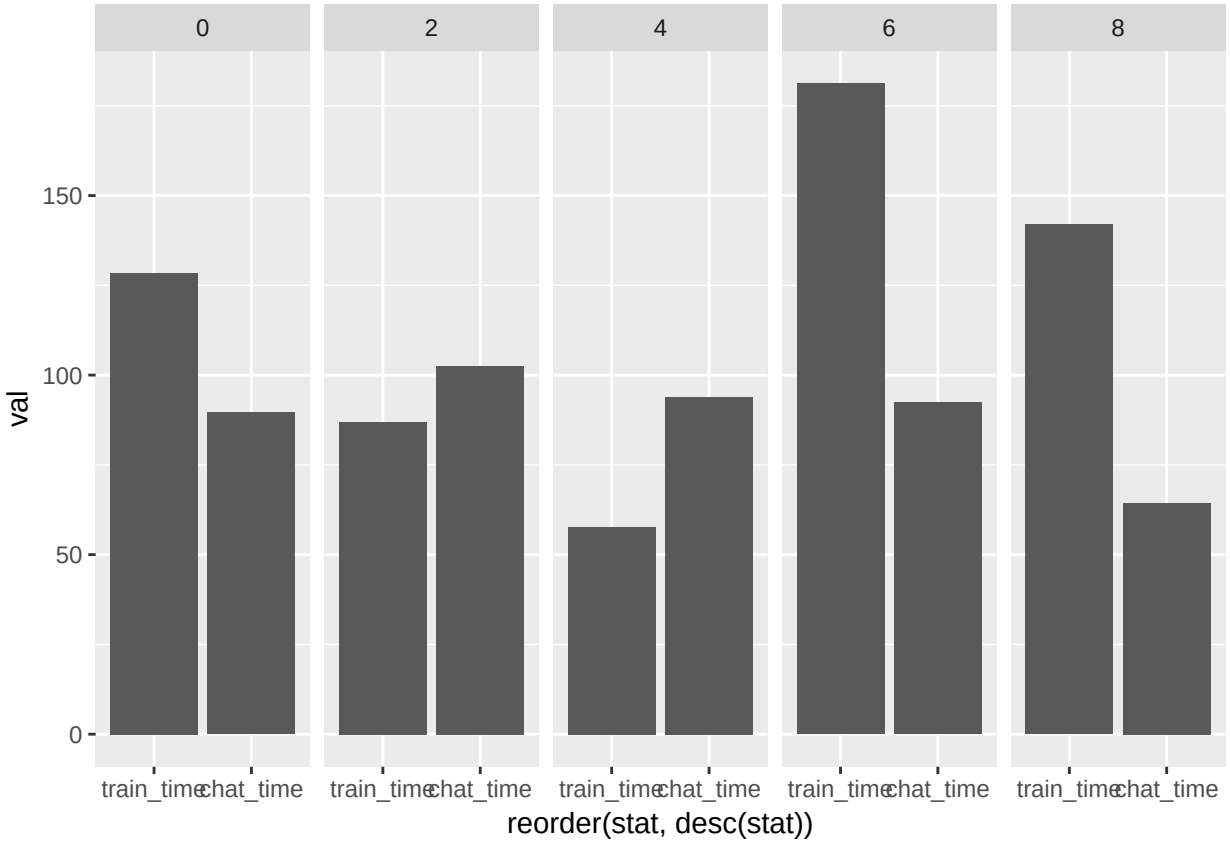
Student versus Teacher Performance – Concepts (Summurization)



Student versus Teacher Performance – Concept Types (Summurization)



Time Comparisons (Summarization)



For Simple Rules 0, 2, 4 (Single Feature & Conjunction) we see that there isn't a wide variance in terms of train time and chat time; but this rift becomes noticeable when we have Harder Rules 6, 8 (Disjunction).